Universidade de Aveiro
2011

Departamento de Electrónica,
Telecomunicações e Informática

Susana Maria Borges Matias de Abreu e Vasconcelos

Ferramentas para processamento de distâncias inter-simbólicas no ADN

Universidade de Aveiro
2011

Departamento de Electrónica,
Telecomunicações e Informática

Susana Maria Borges Matias de Abreu e Vasconcelos

Ferramentas para processamento de distâncias inter-simbólicas no ADN

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Carlos Alberto da Costa Bastos, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e da Doutora Vera Mónica Almeida Afreixo, Professora Auxiliar do Departamento de Matemática da Universidade de Aveiro

Aos meus pais.

o júri

presidente

Prof. Doutor Armando José Formoso de Pinho
Professor Associado com Agregação da Universidade de Aveiro

vogais

Prof. Doutor Carlos Alberto da Costa Bastos
Professor Auxiliar da Universidade de Aveiro (Orientador)

Profª Doutora Vera Mónica Almeida Afreixo
Professora Auxiliar da Universidade de Aveiro (Co-orientadora)

Prof. Doutor José Paulo Ferreira Lousado
Professor Adjunto da Escola Superior de Tecnologia e Gestão de Lamego do Instituto Politécnico de Viseu

agradecimentos

Agradeço à Professora Vera e ao Professor Carlos pela orientação que me deram neste trabalho e por terem estado sempre disponíveis a ajudar quando foi necessário.

Um obrigado muito especial às minhas irmãs, ao Pedro e aos amigos impecáveis que tenho, por me apoiarem sempre e me acompanharem nos melhores e piores momentos da minha vida.

Por último, quero agradecer aos meus pais por tudo. A eles dedico este trabalho.

palavras-chave Bioinformática, ADN, Oligonucleótidos, Distâncias entre símbolos

resumo

A sequenciação do primeiro genoma abriu as portas para o desafio de descobrir mecanismos que permitam estudar e compreender a estrutura do ADN. Com a evolução das técnicas de sequenciação dos últimos quarenta anos, foram geradas grandes quantidades de informação genética, o que levou à necessidade de criar ferramentas computacionais que facilitem a sua análise. Diferentes estudos têm sido realizados com o objectivo de descobrir novos padrões genéticos e tentar compreender a relação entre várias espécies, sendo frequente o uso de mapeamentos que descrevem sequências de ADN e contribuem para a análise das mesmas.

O objectivo desta dissertação consiste em explorar o mapeamento de distâncias entre oligonucleótidos, através da criação de uma ferramenta de suporte a este estudo. A ferramenta desenvolvida permite uma análise comparativa de sequências, utilizando vários métodos quantitativos e proporcionando uma visualização gráfica dos mesmos. Possibilita não só o processamento integral de vários ficheiros, mas também a selecção de uma zona particular do ficheiro a processar. De maneira a tornar a sua utilização mais intuitiva, foi ainda construída uma interface gráfica.

Depois de terem sido realizados alguns estudos com o auxílio da ferramenta desenvolvida, verificou-se que este mapeamento permite detectar padrões genéticos que constituem características distintas de cada espécie.

keywords Bioinformatics, DNA, Oligonucleotides, Distances between symbols

abstract The first genome sequencing led to the challenge of discovering mechanisms that allow to study and to understand the DNA structure. With the evolution of sequencing techniques of the last 40 years, large amounts of genetic data have been generated, leaving the necessity of creating computer tools to facilitate its analysis. Different studies have been carried out to find new genetic patterns and to try to understand the relations between different species, being common the use of mappings to describe and analyze DNA sequences.
The goal of this dissertation consists on exploring the mapping of distances between oligonucleotides, by creating a tool to support this study. The developed tool allows a comparative analysis of sequences, using different quantitative methods and providing graphical visualizations. It enables not only the full processing of several files but also the selection of a particular region of the file to process. In order to make its use more intuitive, a graphical interface was also built.
After carrying out some studies with the help of the developed tool, it was verified that the distance mapping allows the detection of genetic patterns that constitute a distinctive characteristic of each species.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

A	Adenina
ADN	Ácido DesoxirriboNucleico
API	*Application Programming Interface*
ARN	Ácido Ribonucleico
ARNm	Ácido Ribonucleico Mensageiro
ARNt	Ácido Ribonucleico de Transferência
C	Citosina
DDBJ	*DNA Data Bank of Japan*
ddATP	*Dideoxyadenosine Triphosphate*
ddCTP	*Dideoxycytidine Triphosphate*
ddGTP	*Dideoxyguanosine Triphosphate*
ddTTP	*Dideoxythymidine Triphosphate*
EDT	*Event Dispatch Thread*
EMBL	*European Molecular Biology Laboratory*
FTP	*File Transfer Protocol*
G	Guanina
GenBank	*Genetic Sequence Databank*
i.i.d.	independente e igualmente distribuído
INSDC	*International Nucleotide Sequence Database Collaboration*
JVM	*Java Virtual Machine*
K	Grupo ceto (bases G e T)
M	Grupo amino (bases A e C)
NAR	*Nucleic Acids Research*

R	Purina (bases G e A)
S	Ligações fortes de hidrogénio (bases G e C)
T	Timina
TIGR	*The Institute for Genomic Research*
UML	*Unified Modeling Language*
W	Ligações fracas de hidrogénio (bases A e T)
WGS	*Whole Genome ShotGun*
Y	Pirimidina (bases T e C)

Glossário

Ácidos Nucleicos Moléculas de grandes dimensões, geralmente encontradas no núcleo das células e/ou citoplasma, feitas de bases nucleotídicas;

Aminoácidos Constituintes fundamentais das proteínas. São substâncias orgânicas que apresentam na sua estrutura o grupo carboxílico e o grupo amina;

Clustering É uma técnica de *Data Mining* que tem como objectivo criar agrupamentos automáticos de dados (*cluster*) segundo um grau de semelhança;

Codão É a unidade básica do código genético constituída por uma sequência de três nucleótidos (tripleto) do Ácido Ribonucleico Mensageiro (ARNm), que caracteriza um determinado aminoácido ou um sinal de terminação;

Complemento invertido Refere-se a sequências de nucleótidos. Obtém-se através da troca de cada nucleótido pela sua base complementar e posterior inversão da sua ordem. Por exemplo, a sequência complementar de $GACTGC$ é $CTGACG$, invertendo a sequência obtém-se $GCAGTC$;

Cromossoma Sequência de ADN que contém os factores de hereditariedade responsáveis pela codificação da informação genética;

Dinucleótido É um fragmento de uma cadeia simples de ácido nucleico (ADN ou ARN) constituído por dois nucleótidos;

Electroforese É uma técnica de separação de moléculas que consiste na migração de moléculas com carga, numa solução, em função da aplicação de um campo eléctrico. Aplica-se no campo da bioquímica na separação de compostos que possuem carga (aminoácidos, péptidos, proteínas, ácidos nucleicos) tendo em conta que a carga destas substâncias depende do pH do meio em que se encontram;

Genoma É o conjunto completo de cromossomas existentes num organismo;

Javadoc É um gerador de documentação criado pela *Sun Microsystems* para documentar programas em Java a partir do código-fonte. O resultado expresso em *HTML* é constituído por algumas marcações muitos simples inseridas nos comentários do programa;

Nucleótido Unidade básica do ADN ou ARN, constituída por uma pentose, um grupo fosfato e uma base nitrogenada;

Oligonucleótido É um fragmento curto de uma cadeia simples de ácido nucleico (ADN ou ARN), tipicamente com 20 ou menos bases;

Reading Frame Em biologia, uma *reading frame* consiste numa maneira de partir uma sequência de nucleótidos (ADN ou ARN) em codões, constituídos por três letras, que podem ser traduzidos em aminoácidos. Neste contexto, existem três possíveis *reading frames*, sendo que cada uma começa num alinhamento diferente. A definição de *reading frame* poderá também ser usada num outro contexto, onde as sequências não se partem em grupos de três letras mas em grupos de N letras. Neste caso, existirão N possíveis *reading frames*;

Spinner É um elemento usado em interface gráfica que consiste numa caixa de texto em que o utilizador pode ajustar um valor, aumentando-o e diminuindo-o através do uso de setas;

Thread Em ciências da computação um *thread* resulta da divisão do fluxo de execução de um processo em duas ou mais tarefas concorrentes;

Trinucleótido É um fragmento de uma cadeia simples de ácido nucleico (ADN ou ARN) constituído por três nucleótidos;

Tooltip É um elemento bastante comum em interface gráfica usado em conjunto com um *cursor*, em que o utilizador passa o *cursor* por cima de um elemento gráfico e surge uma caixa de texto com informação sobre esse elemento;

Capítulo 1

Introdução

1.1 Enquadramento e motivação

Desde há muito tempo que se tenta compreender de que forma as características que distinguem as diferentes espécies são passadas de geração em geração. A semelhança de uma criança com os seus pais sempre foi evidente, mas não existia um método científico que determinasse de que forma eram herdadas essas características. Em 1865 *Gregor Mendel* [22] descobriu que os traços individuais são determinados por diferentes factores, mais tarde baptizados de genes.

A base biológica manteve-se desconhecida até que foi descoberto em 1944 que o gene era composto por Ácido DesoxirriboNucleico (ADN). Cerca de dez anos mais tarde, *James Watson e Francis Crik* [40] propuseram a agora famosa estrutura de dupla hélice do ADN. Este modelo foi uma contribuição vital para o aprofundamento da compreensão da hereditariedade.

Consequentemente, diversas tecnologias foram desenvolvidas e houve um forte investimento no que diz respeito a técnicas de sequenciação, tendo sido feita a primeira sequenciação completa de um organismo em 1977. Desde então, o estudo da estrutura e do funcionamento do ADN tornou-se num dos tópicos mais estudados em Biologia Molecular.

Nos últimos quarenta anos, houve uma grande evolução das técnicas de sequenciação o que permitiu que actualmente existam já centenas de genomas sequenciados. Evidentemente, a grande quantidade de informação fornecida por estas tecnologias fez com que surgissem novos desafios, nomeadamente descobrir formas de organizar e analisar os dados obtidos.

O facto de grande parte da investigação em Biologia Molecular envolver aplicações matemáticas e estatísticas, em particular operações repetitivas e matematicamente complexas, tornou inevitável a união da Informática com a Biologia, dando origem ao ramo da Bioinformática. A Bioinformática define-se então como sendo toda a tecnologia que usa computadores para armazenamento, recuperação, manipulação e distribuição de informação relativa a moléculas tais como o ADN, Ácido Ribonucleico (ARN) e proteínas [42]. Pode ser ainda dividida em dois campos específicos: o desenvolvimento de ferramentas computacionais e de bases de dados, e a aplicação das mesmas de maneira a gerar conhecimento biológico para melhor entender os seres vivos.

As ferramentas computacionais são usadas em três ramos da investigação em Biologia Molecular e Genética: análise de sequências moleculares, análise de estruturas moleculares e análise de funções moleculares [42]. A área de análise de sequências inclui o alinhamento de sequências, a sua pesquisa em bases de dados, a descoberta de padrões e repetições, a pesquisa

de genes, a reconstrução de relações evolucionárias e a comparação de genomas. A análise estrutural engloba a análise da estrutura de proteínas e nucleótidos, assim como a comparação, a classificação e a previsão dos mesmos. Por último, a análise funcional abrange o perfil de expressão de genes, a previsão da interacção proteína-proteína, a previsão da localização de proteínas subcelulares, reconstrução do caminho metabólico e a sua simulação.

Sem dúvida que a Bioinformática é uma área de grande potencial que revolucionou a pesquisa biológica ao longo das últimas décadas e que continua em grande expansão. Diversas soluções que visam analisar sequências genéticas têm sido criadas, focando-se nos diferentes tipos de análise acima descritos.

Nalguns estudos, é frequente converter a sequência genética numa sequência numérica através de um processo de mapeamento. Este trabalho irá centrar-se no mapeamento que converte a sequência genética numa sequência numérica correspondente às distâncias entre símbolos iguais [1, 4]. Esta metodologia revelou ser útil por caracterizar sequências de ADN e consequentemente detectar padrões genéticos que constituem características distintas de cada espécie.

Assim, surgiu o interesse de explorar esta metodologia através da construção de uma ferramenta computacional que permita utilizar o mapeamento das distâncias entre símbolos. O desenvolvimento da ferramenta pretende facilitar o estudo deste mapeamento, através da utilização de diferentes análises quantitativas e criação de gráficos, podendo resultar em informações úteis para aprofundar o conhecimento sobre a estrutura genética dos seres vivos e, consequentemente, a relação entre eles.

1.2 Objectivos

O principal objectivo deste trabalho consiste em desenvolver uma ferramenta que auxilie o estudo da sequência de distâncias inter-simbólicas de uma forma simples e eficiente.

Esta ferramenta deverá converter uma sequência genética, composta por letras, numa sequência numérica baseada nas distâncias entre símbolos. Símbolos estes que podem ser compostos por uma ou mais letras. A sequência resultante será então usada para efectuar análises quantitativas e criar diferentes tipos de visualização gráfica que possibilitem uma melhor interpretação dos resultados. Com o objectivo de criar uma ferramenta intuitiva e de fácil utilização, será também desenvolvida uma interface gráfica baseada em paradigmas familiares ao utilizador. De uma forma mais específica, para a concretização deste trabalho foram estabelecidas as seguintes tarefas:

- estudar e compreender as noções básicas de biologia molecular e computacional;
- pesquisar e estudar diferentes processos de mapeamento existentes;
- estudar o mapeamento das sequências de distâncias inter-simbólicas;
- desenvolver uma ferramenta que permita o estudo do comportamento das distâncias de um genoma ou cromossoma completo (análise global) e de apenas uma parte do cromossoma, escolhida pelo utilizador (análise local);
- implementar uma interface gráfica para a ferramenta;
- realizar testes de desempenho.

1.3 Estrutura da dissertação

Este documento está dividido em cinco capítulos e um anexo.

No presente capítulo é feito um enquadramento histórico sobre o tema da dissertação e são também descritos os objectivos gerais e a estrutura da mesma.

No capítulo dois será feita uma introdução à Biologia Molecular, onde serão descritas algumas das principais características do ADN, conhecimentos fundamentais para o desenvolvimento deste trabalho. Serão também apresentadas algumas técnicas de sequenciação do ADN, assim como vários métodos de representação do mesmo, que se dividem em métodos gráficos e representações numéricas, sendo dada especial ênfase à representação das distâncias inter-simbólicas. Serão ainda descritas algumas ferramentas existentes no âmbito da análise de sequências de ADN e apresentados os diferentes modelos de dados.

O capítulo três abrangerá todo o processo de desenvolvimento da ferramenta, começando por uma análise dos requisitos funcionais e não funcionais, seguindo-se a definição dos casos de uso da ferramenta e ainda a descrição do diagrama de classes e das estruturas de dados utilizadas. Serão também descritas as diferentes etapas da implementação da ferramenta. Em primeiro lugar, serão apresentadas a linguagem e ferramentas usadas, assim como os motivos das suas escolhas. De seguida, será feita uma descrição detalhada dos modelos de análise, métodos quantitativos e de visualização de dados. Serão ainda descritos alguns algoritmos usados e também o tipo de resultados gerados. Por último, será feita uma apresentação da interface gráfica e do manual de utilizador da ferramenta.

No capítulo quatro serão apresentados alguns exemplos de utilização da ferramenta. Serão então processados genomas completos com oligonucleótidos de vários tamanhos utilizando as duas abordagens de processamento implementadas e será feita uma análise comparativa de diferentes espécies. Para terminar, serão efectuados testes de desempenho, que incluirão várias análises de genomas e cromossomas de diversas espécies e tamanhos, por forma a estimar tempos médios de processamento.

Por último, o capítulo cinco apresentará uma conclusão do trabalho realizado, assim como algumas propostas e ideias para um trabalho futuro.

No final deste documento encontra-se ainda um anexo que contém o manual de utilização da ferramenta desenvolvida.

Capítulo 2

Análise de sequências de ADN

A possibilidade de analisar sequências de ADN deve-se a um intensivo esforço por parte de muitos cientistas, que durante anos procuraram descobrir mecanismos que permitissem sequenciar o ADN. Uma vez atingido esse objectivo, surgiu o desafio de encontrar formas de analisar os dados obtidos, tendo sido criadas diversas representações do ADN para facilitar a sua análise.

Neste capítulo pretende-se contextualizar a temática deste trabalho, introduzindo para isso alguns fundamentos essenciais de Biologia Molecular, abordando o processo de sequenciação, diversas representações do ADN e alguns exemplos de ferramentas que permitem analisar o mesmo. Mais concretamente, será explicado no que consiste o processo de sequenciação do ADN e será fornecida uma perspectiva histórica sobre os métodos de sequenciação existentes. De seguida, serão apresentadas algumas representações do ADN, dividindo-se em métodos gráficos e numéricos, dando especial atenção à representação de distâncias inter-simbólicas em que se baseia este trabalho. Serão ainda mostradas algumas ferramentas computacionais de análise do ADN, abrangendo diferentes áreas de estudo. Serão também mencionadas as bases de dados genéticas mais conhecidas a nível mundial, necessárias para adquirir os dados utilizados neste trabalho e serão ainda descritos os formatos de ficheiros de sequências genéticas mais comuns.

2.1 Fundamentos de Biologia Molecular

O ADN é o suporte molecular da informação biológica que define as características de cada organismo. É composto por pequenas moléculas chamadas nucleótidos, que por sua vez são formadas por três partes: uma pentose, um grupo fosfato e uma base nitrogenada. Os nucleótidos podem ser distinguidos pelas quatro bases existentes: Adenina (A), Citosina (C), Guanina (G) e Timina (T). As bases nitrogenadas podem ser classificadas de acordo com a sua estrutura, sendo a Adenina e a Guanina purinas e a Citosina e a Timina pirimidinas.

Olhando para o ADN como uma linguagem, pode-se dizer que esta é escrita num código que utiliza um alfabeto químico de quatro letras. Letras essas que são combinadas numa ordem particular para formar "palavras", "frases" e "parágrafos" todos ligados entre si criando longas cadeias de ADN [6].

O ADN existente em cada célula é como um livro de receitas, dividido num determinado número de "capítulos", os cromossomas. Cada capítulo contém um número individual de receitas, os genes, e cada gene tem as instruções necessárias para fazer um produto específico.

Assim as mesmas quatro letras deste alfabeto podem ser combinadas de muitas maneiras diferentes para escrever diferentes traços de um organismo [35].

Segundo o modelo de dupla hélice proposto por Watson e Crick [40], a molécula de ADN é constituída por duas cadeias de nucleótidos dispostas em espiral ao redor de um eixo imaginário (Figura 2.1). As duas cadeias unem-se através de pontes de hidrogénio, que se estabelecem entre pares de bases complementares: A com T e C com G. Para além disso, as duas cadeias são antiparalelas, pois têm polaridades opostas. Numa das cadeias as ligações estão orientadas do carbono 3' de um nucleotídeo ao carbono 5' do nucleotídeo adjacente, enquanto que na cadeia complementar a orientação é inversa, do carbono 5' ao 3'.

O ADN contêm não só informações relativas a características hereditárias mas também informação necessária para a produção de proteínas e consequente sobrevivência dos seres vivos. Existem duas partes constituintes do ADN, a região codificante que consiste no conjunto de nucleótidos com significado em termos de produção de proteínas, e as regiões não codificantes.

A cada unidade proteica chama-se aminoácido e o código código de um aminoácido é escrito através de palavras de três nucleótidos, chamados codões. Existem sessenta e quatro codões distintos (4^3) e vinte tipos diferentes de aminoácidos. Diferentes codões podem codificar o mesmo aminoácido. Quando se juntam os tipos correctos de aminoácidos na quantidade e na ordem certa é sintetizada uma proteína.

Assim, para traduzir a mensagem do ADN em proteínas os codões são primeiro copiados para uma molécula semelhante ao ADN, o ARN. Depois desta replicação, o Ácido Ribonucleico Mensageiro (ARNm) transporta informação sobre a sequência da proteína para os ribossomas onde é feita a sua síntese.

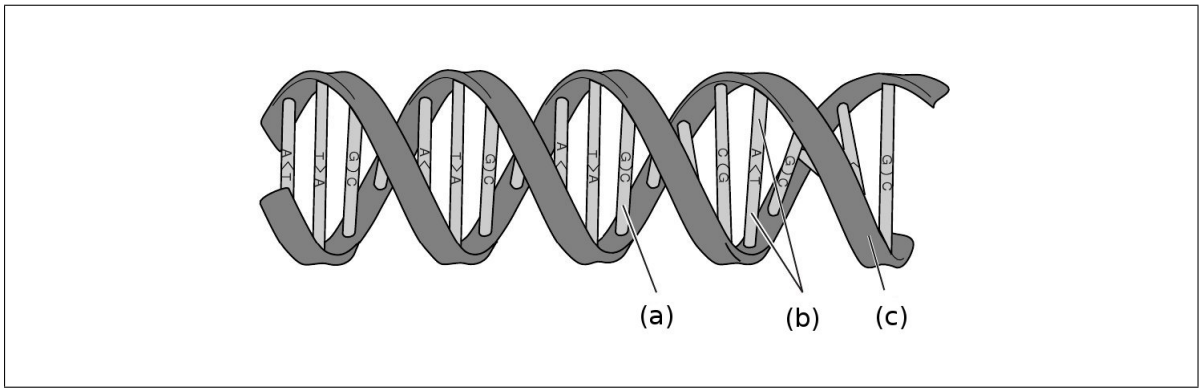


Figura 2.1: Modelo de dupla hélice do ADN com a sua base nitrogenada (a), os pares de bases complementares (b) e a estrutura de açúcar fosfato (c). (Figura adaptada de *http://www.accessexcellence.org/RC/VL/GG/*)

2.2 Sequenciação do ADN

A sequenciação do ADN consiste no processo bioquímico de determinar a ordem dos nucleótidos numa sequência genética.

Os primeiros métodos para sequenciar ácidos nucleicos surgiram em 1965 mas eram aplicáveis apenas em moléculas de ARN, devido essencialmente ao facto destas moléculas serem muito menores que as moléculas de ADN [5]. Nos anos seguintes, o desenvolvimento de métodos de sequenciação do ADN tornou-se numa prioridade para os biólogos, tendo surgido

em 1977 duas técnicas igualmente eficazes: o método de *Sanger* [31] e o método de *Maxam-Gilbert* [21].

O método original de *Sanger*, conhecido também por método de sequenciação de terminadores de cadeia, baseia-se na síntese enzimática e foi usado em 1977 para sequenciar o primeiro genoma, o *Bacteriophage ϕ X74*, um genoma viral com apenas 5.368 bases [29]. Este método utiliza nucleótidos terminadores (ddATP, ddCTP, ddGTP, ddTTP) para determinar os nucleótidos da sequência, sendo então preparadas quatro reacções de sequenciação e em cada uma incorporada uma espécie de terminador. Desta forma, cada reacção revela a posição em que aparece o respectivo nucleótido. A electroforese destas quatro reacções de sequenciação em paralelo permite então obter a ordem em que estão dispostos os nucleótidos (Figura 2.2a).

O método de *Maxam-Gilbert* tem como princípio básico a decomposição química da molécula de ADN causando a divisão da mesma de uma forma específica (por exemplo, em G, A+G,C+T e C). Cada reacção ocorre num tubo diferente, para que de seguida os fragmentos resultantes possam ser analisados através da electroforese (Figura 2.2b).

Os princípios básicos destes dois métodos continuaram a ser usados mas com significativas modificações face os avanços tecnológicos das últimas décadas. No entanto, o método de *Sanger* tornou-se no mais popular, já que o método de *Maxam-Gilbert* tem uma grande complexidade técnica e torna-se mais difícil de usar para moléculas muito extensas.

Com o objectivo de acelerar o processo de sequenciação, foi desenvolvida uma estratégia alternativa, o *Whole Genome ShotGun* (WGS). Ela consiste em dividir aleatoriamente a sequência de ADN em vários fragmentos de maneira a obter múltiplas sobreposições, sequenciar cada fragmento e remontar a sequência com base no alinhamento e nas sobreposições da sequência. Uma vez que esta estratégia é propensa a erros, os grupos de investigação demoraram algum tempo até a adoptarem, optando por abordagens baseadas em mapeamentos mais organizados. Mas em 1983 *Sanger* utilizou com sucesso esta técnica [30] e a partir daí, os projectos de sequenciação avançaram mais depressa e outros genomas virais foram sequenciados.

Em 1995, Craig Venter, Hamilton Smith e colegas do *The Institute for Genomic Research* (TIGR) publicaram o primeiro genoma completo de um organismo vivo, *Bacterium Haemophilus Influenzae* [9], com um cromossoma circular composto por 1.830.137 bases. Esta publicação marcou a primeira sequenciação de um genoma completo usando a abordagem WGS.

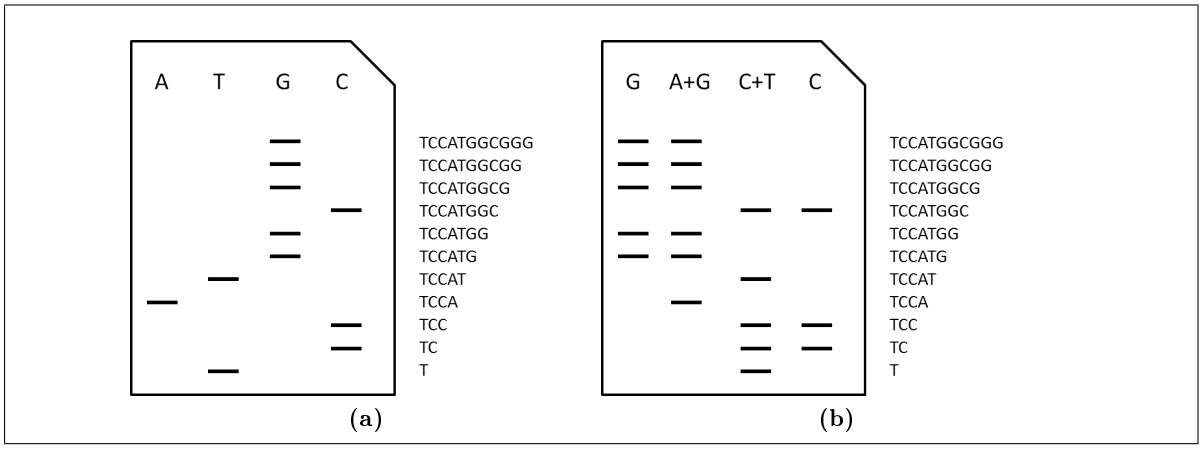


Figura 2.2: Método de sequenciação de *Sanger* (a) e *Maxam-Gilbert* (b). (Figuras adaptadas de [5])

Finalmente, em 2003 foi concluída a sequenciação do genoma humano completo com aproximadamente três mil milhões de letras genéticas, treze anos depois de ter sido criado o *Human Genome Project* [7].

Nos anos que se seguiram ao projecto de sequenciação do genoma humano, surgiram a segunda [20] e terceira geração de métodos de sequenciação [32]. Estes métodos vieram reduzir significativamente o tempo e os recursos necessários para sequenciar genomas. Consequentemente, nos últimos anos verificou-se uma tendência constante na redução de custos deste tipo de serviço, tornando a possibilidade de qualquer pessoa ter o seu genoma sequenciado uma realidade.

2.3 Representação do ADN

A grande quantidade de dados genéticos tornou necessária a criação de técnicas de mapeamento do ADN para analisar e extrair informação relevante. Nos últimos anos, surgiram diferentes métodos para interpretar sequências de ADN com o objectivo de quantitativamente comparar sequências e determinar semelhanças entre elas.

Nesta secção serão descritos alguns desses métodos, começando por representações gráficas que proporcionam pistas visuais de bases dominantes em diferentes regiões da sequência e podem ser usadas para identificação de genes. Seguindo-se abordagens que descrevem numericamente as sequências de ADN, permitindo capturar a essência da composição e da distribuição das bases de uma sequência de uma maneira quantitativa.

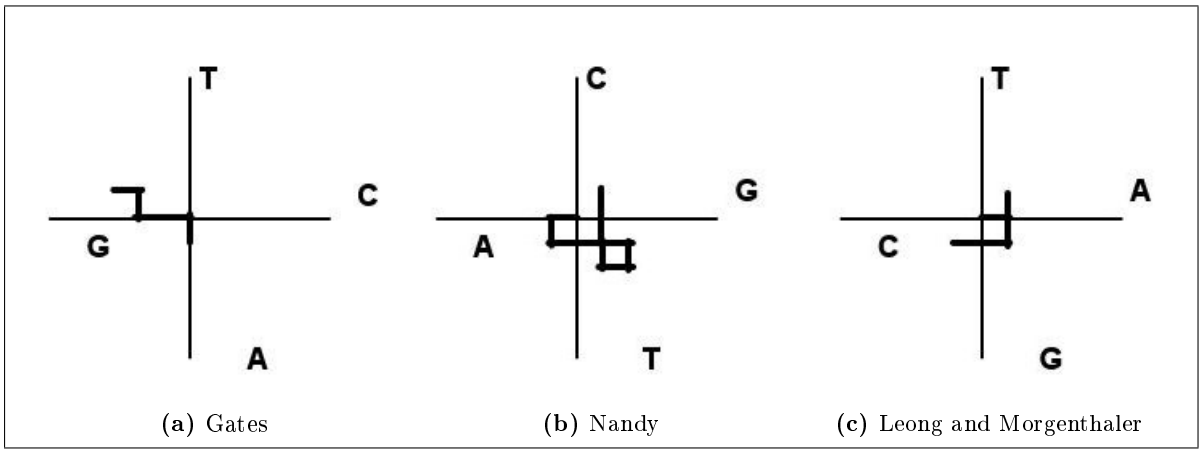
Por último, será descrito o mapeamento das distâncias inter-simbólicas que servirá de base para o desenvolvimento deste trabalho.

2.3.1 Métodos gráficos

Muitas técnicas têm sido usadas para representar graficamente sequências de ADN, desde simples métodos de duas dimensões a métodos muito complexos com três dimensões ou mais. Como veremos, métodos que utilizam várias dimensões muitas vezes tornam-se difíceis de visualizar e, por outro lado, métodos com apenas duas dimensões que não excluem caminhos repetidos perdem parte da informação. Desta forma, para saber qual a melhor escolha é crucial entender as vantagens e desvantagens de cada método e acima de tudo o tipo de informação que ele fornece.

O uso das quatro direcções cardeais para representar as quatro bases nucleicas num gráfico de sequências de ADN foi implementado pela primeira vez por Gates, em 1985 [11]. Neste tipo de representação é atribuída uma direcção e uma distância a cada base. Assim, ao ler uma sequência de ADN é desenhado um novo ponto de acordo com a direcção associada à base que vai surgindo, criando desta forma um caminho que salienta elementos estruturais da composição dos nucleótidos.

Existem diferentes esquemas de representação, um redescoberto independentemente por *Nandy* em 1994 [28] e outro por *Leong* e *Morgenthaler* em 1995 [18], que variam apenas na direcção associada a cada base. Por exemplo, de acordo com a descrição de *Nandy*, um ponto é desenhado avançando uma posição na direcção negativa de x se a base for Adenina, na posição oposta se for Guanina, avançando uma posição na direcção positiva de y se a base for Citosina e na oposição oposta se for Timina. Como podemos ver na Figura 2.3, estas representações contêm limitações para sequências com bases alternadas, como por exemplo $GAGAGAGAG$,



(a) Gates **(b)** Nandy **(c)** Leong and Morgenthaler

Figura 2.3: Exemplos de esquemas de representação 2D para a sequência $ATGGTGCACC$

causando caminhos sobrepostos ao longo do eixo dos x e consequentemente conduzem a uma perda de informação.

De maneira a evitar sobreposições como as verificadas nas representações anteriores, *He* e *Wang* [15] definiram outra caracterização que divide as quatro bases nos seus grupos estruturais. Elas podem ser divididas em duas classes de acordo com a sua estrutura química: em purina ($R = (A, G)$) e pirimidina ($Y = (C, T)$), ou grupo amino ($M = (A, C)$) e grupo ceto ($K = (G, T)$). Para além disso, a divisão pode também ser feita de acordo com a força da ligação de hidrogénio, ou seja, ligação fraca ($W = (A, T)$) e ligação forte ($S = (G, C)$). Assim, cada sequência é desenhada de acordo com estas coordenadas características, dando-se o nome de gráficos característicos. Desta forma, elimina-se a sobreposição e melhora-se a análise visual das diferentes características estruturais e das ligações de uma sequência. Este método acabou por ser usado e aprofundado por *Song* e *Tang* [36], entre outros autores. Na Figura 2.4 podemos ver dois dos doze gráficos possíveis de criar utilizando esta representação.

Em 2003, *Wu et al.* [41] apresentaram a *DB-Curve (Dual-Base Curve)* que mostrava duas das quatro bases ao mesmo tempo no plano. A ideia desta representação é que, se duas sequências são semelhantes, esta semelhança também deverá ser reflectida nas suas sub-sequências, constituídas por duas das suas quatro bases. Tomando como exemplo a *AC DB-Curve*, atribuem-se os seguinte vectores às respectivas bases:

$$\begin{aligned} (1,1) &\implies A \\ (-1,1) &\implies C \\ (0,1) &\implies \text{T e G} \end{aligned}$$

Considerando o ponto de partida $(0, 0)$, uma sequência de ADN pode ser mapeada num sistema coordenado de duas dimensões por um gráfico cumulativo das bases da sequência usando a notação descrita. Como podemos ver na Figura 2.5b, a *AC DB-Curve* acentua as relações entre as bases A e C, tornando a sua visualização simples e clara. Outras cinco *DB-Curve* podem ser obtidas do mesmo modo para AG, AT, TC, CG e TG.

Baseando-se também no uso de vectores, *Zhu-Jin Zhang* [44] sugeriu a utilização de uma *DV-Curve (Dual-Vector Curve)*. Esta representação, para além de evitar perda de informação, permite uma boa visualização independentemente do tamanho da sequência. Como representado na Figura 2.6a, a cada base A, T, C e G são associados dois vectores da seguinte forma:

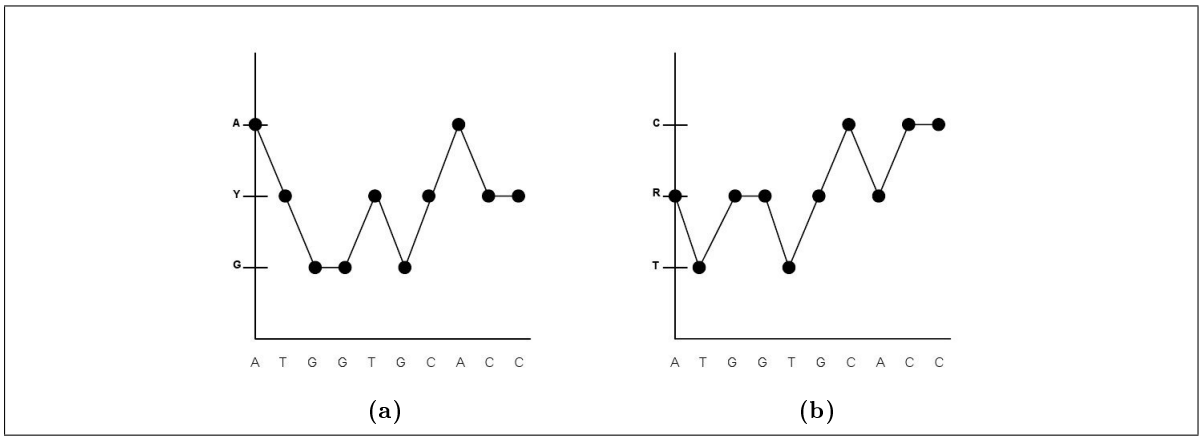


Figura 2.4: Representação 2D das curvas de Pirimidina (bases T e C) (a) e Purina (bases G e A) (b) da sequência ($ATGGTGCACC$) de *Song* e *Tang* [36].

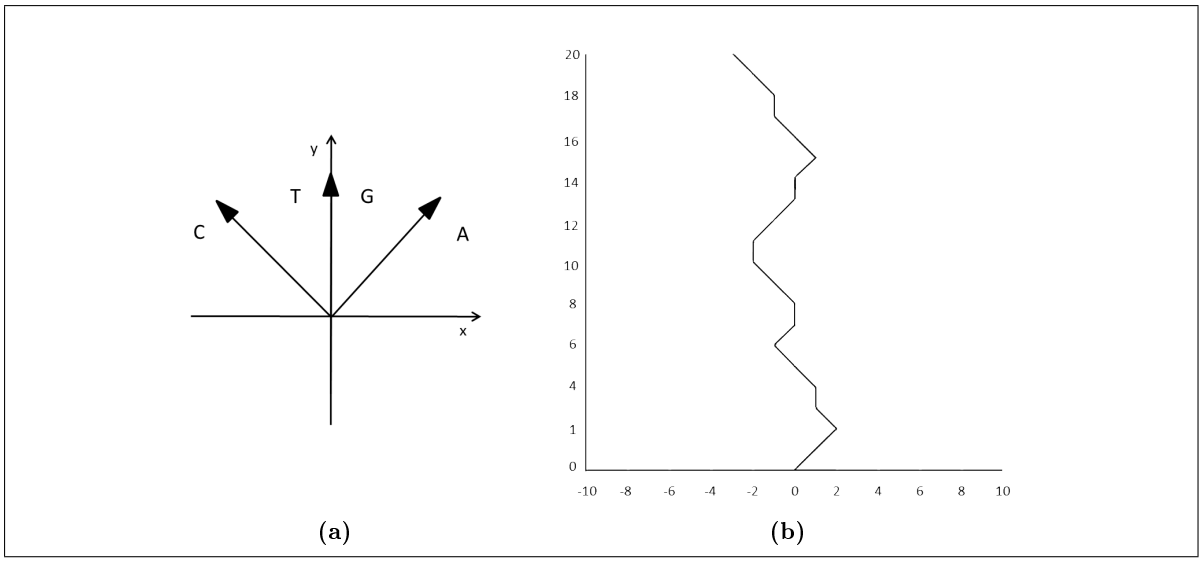


Figura 2.5: Representação gráfica da *AC DB-Curve* de *Wu et al.* [41]. Vectores associados a cada base (a) e *AC DB-Curve* da sequência $AACGCCATCCGAAGACCTCC$ (b).

$$\begin{aligned}(1,1),(1,1) &\implies A\\ (1,1),(1,-1) &\implies T\\ (1,-1),(1,1) &\implies C\\ (1,-1),(1,-1) &\implies G\end{aligned}$$

Cada nucleótido ocupa então duas unidades ao longo do eixo dos x (Figura 2.6b). Para além disso, analisando a curva ao longo do eixo dos y conseguimos extrair informação sobre as bases que têm uma maior frequência. De facto, quando os valores ao longo do eixo dos y são superiores a zero significa que existem mais bases do tipo A do que do tipo G e quando os valores são inferiores a zero existem mais bases do tipo G do que A. Este tipo de curvas pode ter diferentes aplicações, desde simples comparações de semelhança a estudos de mutações genéticas.

Outra aproximação 2D usada para representação gráfica, consiste em desenhar quatro

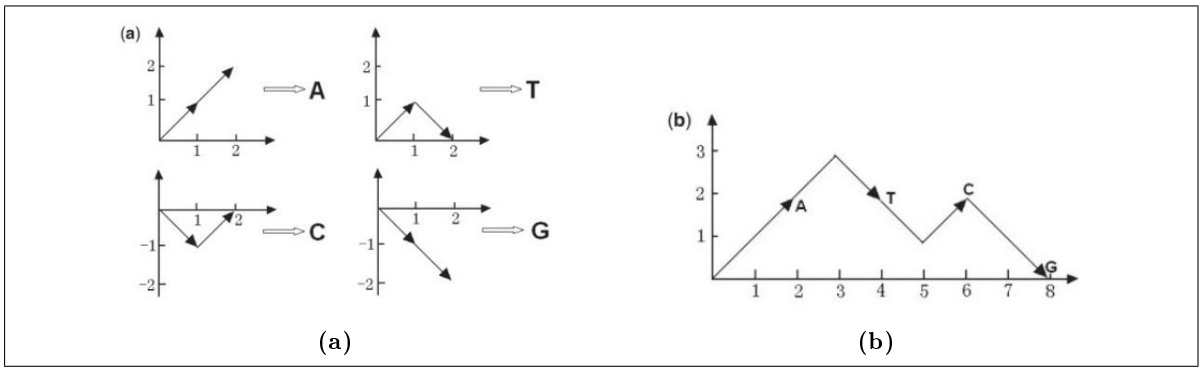


Figura 2.6: Representação gráfica da *DV-Curve* de *Zhu-Jin Zhang* [44]. Vectores associados a cada base (a) e *DV-Curve* da sequência $ATCG$ (b).

linhas horizontais numa superfície e associá-las a cada uma das quatro bases. Por baixo das linhas inserem-se as bases da sequência de ADN a analisar, distanciando-as em uma unidade. De seguida, é posicionado um ponto ao longo da linha horizontal para cada base na sequência e no final unem-se todos os pontos, como mostra a Figura 2.7a. Existem 24 (4!) grafos possíveis tendo em conta que as linhas podem ser legendadas em qualquer ordem e não existe perda de informação. Este método é usado por vários autores, incluindo *Randić, Vračko, Lerš e Plavšić* [25,27].

A representação acima descrita pode também ser feita utilizando um método binário. Neste caso, as bases são separadas nas três classificações relativas ao seu grupo e à sua estrutura química, sendo o valor 1 associado aos valores R, M ou W, e o valor 0 aos tipos Y, K ou S. Desta forma, existirão três grafos característicos para cada sequência de ADN. Assim, a sequência $ATGGTGCACC$ será representada como mostra a Figura 2.7b. Este método foi usado por vários autores, entre eles *Li e Wang* [19].

Outra representação gráfica, usada por *Yao e Wang* [16], utiliza unidades de quadrado intituladas de células, onde os quatro pontos dos cantos são designados como as quatro bases. A coordenada x da base na célula unitária é obtida através da descoberta da coluna onde está a base individual. Numerando a primeira coluna como zero, as colunas pares são determinadas pela fórmula $(2(i-1))$ e as colunas ímpares pela fórmula $((2(i-1))+1)$ onde i é o número da base. A coordenada y é encontrada tendo em conta se está na primeira ou segunda linha da célula. Concluindo, as seguintes descrições são dadas para cada base:

$$\begin{aligned} G &= (0,\ 2(i-1)) \\ A &= (1,\ 2(i-1)) \\ C &= (0,\ 2(i-1)+1) \\ T &= (1,\ 2(i-1)+1) \end{aligned}$$

onde i é a posição da base na sequência. Assim, usando a mesma sequência $ATGGTGCACC$ obtém-se um grafo como o da Figura 2.8a.

Uma abordagem bastante diferente consiste em fazer uma *worm curve* para representar uma sequência. Nesta representação considera-se que cada base está associada a um conjunto de números ($A = 00$, $G = 01$, $C = 10$, $T = 11$). Esses números serão usados para reescrever a sequência, substituindo cada base pelo seu dígito respectivo. Uma vez criada a nova sequência, passa-se ao desenho da curva que envolve séries de linhas verticais e horizontais onde cada linha vertical representa uma base e cada linha horizontal representa a ligação das bases. É

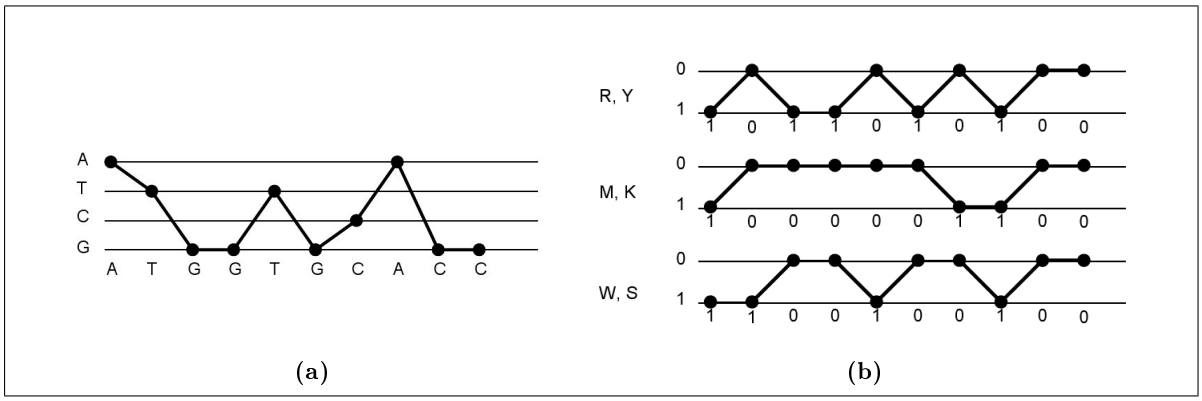


Figura 2.7: Representação gráfica que utiliza quatro linhas horizontais usadas por *Randić et al.* [25,27] (a) e duas linhas horizontais usada por *Li e Wang* [19] (b).

feita uma volta de 90 graus à direita em cada passo e, se esta coincidir com um local já visitado, é feita uma volta de 270 graus no sentido oposto. No final é marcado um ponto em cada canto que equivale ao valor um. Utilizando como exemplo a sequência inicial $ATGGTGCACC$, a sequência equivalente seria:

$$\{00, 11, 01, 01, 11, 01, 10, 00, 10, 10\}$$

Começando no centro da curva, é então feita uma linha vertical que equivale à primeira base (Figura 2.8b). Uma vez que a primeira base tem o valor 00, não é desenhado nenhum ponto nesta recta. De seguida, é feita uma linha 90 graus à direita que liga a primeira à segunda base. A segunda base é representada através de outra linha vertical e é adicionado um ponto ao início e final da recta pois esta tem o valor 11. Como uma deslocação de 90 graus iria coincidir com um local já visitado, é feita uma uma volta de 270 graus no sentido oposto e desenhada a linha que liga a segunda base à terceira. Mais uma vez, é feita outra linha vertical que equivale à terceira base. Como esta base tem o valor 01 é feito um ponto no final da recta. O processo repete-se até se atingir a última posição da sequência. Este método foi usado por usado por *Randić et al.* [26] e tem a vantagem de evitar a intersecção da curva e usar menos espaço que os métodos anteriores.

Uma representação gráfica 3D foi originalmente proposta por *Hamori e Ruskin* [14] em 1983, conhecida como *H-Curve*. Esta curva é construída movendo uma unidade ao longo de uma das quatro direcções, que representam as quatro bases num plano xy, e uma para cada unidade no eixo do z (Figura 2.9). A direcção positiva de z é então usada para contar o número de bases da sequência. Embora sequências de ADN muitos longas possam ser desenhadas usando *H-Curves*, micro características são perdidas à medida que as sequências aumentam. *Randić e Liao* propuseram algumas representação que também evitavam as limitações associadas com a sobreposição mas aumentavam o tempo computacional e o espaço utilizado.

Uma representação distinta em 3D que proporciona uma única representação para visualização e análise de uma sequência de ADN é a *Z-Curve* de *Zhang e Zhang* [43]. Este método tem três componentes (x_n, y_n, z_n) que representam três distribuições independentes de nucleótidos que descrevem completamente a sequência. Eles mostram as distribuições de bases purina versus pirimidina (R versus Y), grupo amina versus ceto (M versus K), e ligações fracas versus ligações fortes (S versus W) ao longo da sequência e representam-se da seguinte forma:

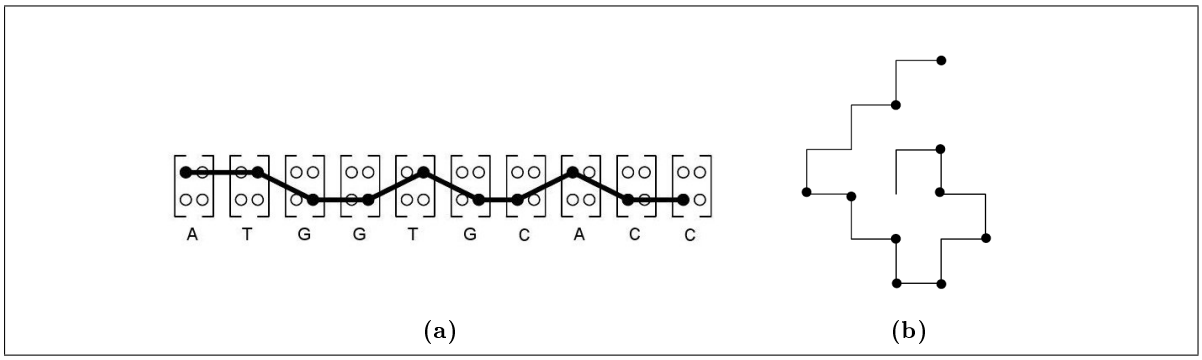


Figura 2.8: Representação gráfica 2D baseada em células de *Yao e Wang* [16] (a) e a *Worm Curve* usada por *Randić et al.* [26] (b).

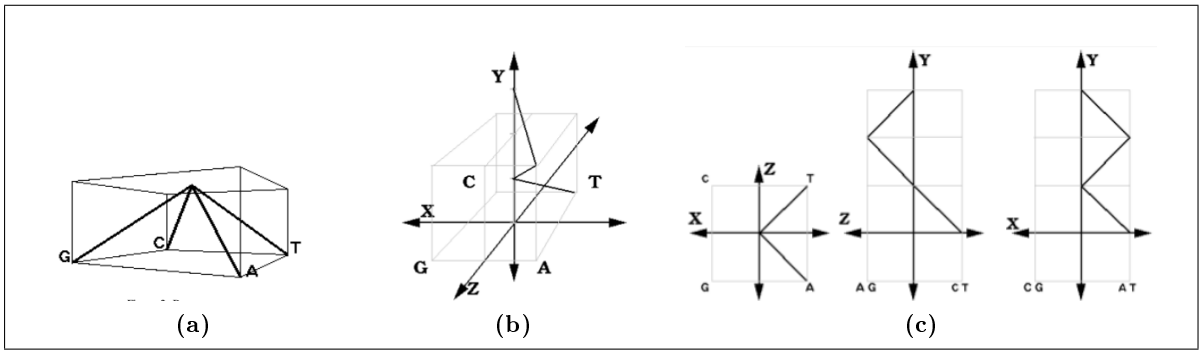


Figura 2.9: *H-Curve* de *Hamori* e *Ruskin* [14]. Vectores associados a cada base (a), exemplo da *H-Curve* para a sequência ACT (b) e as as suas três projecções no plano (b).

$$x_n = (A_n + G_n) - (C_n + T_n) = R_n - Y_n$$

$$y_n = (A_n + C_n) - (G_n + T_n) = M_n - K_n$$

$$z_n = (A_n + T_n) - (C_n + G_n) = W_n - S_n$$

$$(n = 0, 1, 2, ..., N, x_n, y_n, z_n \in [-N, N])$$

onde A_n, C_n, G_n e T_n são o número total de ocorrências de A, C, G e T da subsequência constituída pelas bases do índice 0 ao índice n. Assim, para a sequência $CGAT$ os componentes tomariam os seguintes valores:

$$x_n = [-1, 0, 1, 0]$$

$$y_n = [1, 0, 1, 0]$$

$$z_n = [-1, -2, -1, 0]$$

A curva 3D pode também ser representada a duas dimensões, baseada na disparidade de RY e MK, na disparidade de AT e GC, ou ainda representada por um dos componentes da *Z-curve* ao longo do cromossoma (Figura 2.10). Assim, com esta representação é possível reconhecer mais facilmente alguns padrões no que diz respeito à composição da sequência.

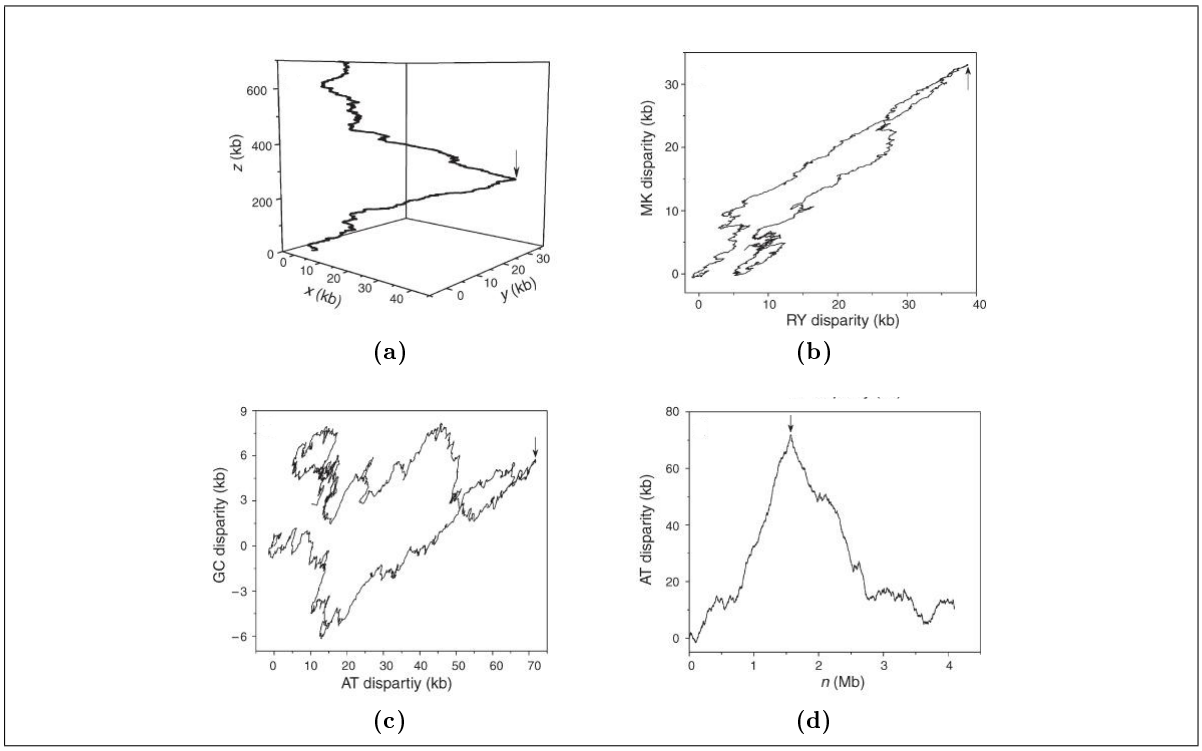


Figura 2.10: Representação gráfica 3D da *Z-Curve* de *Zhang e Zhang* [43] do genoma *Methanosarcina mazei*. A *Z-Curve* 3D (a), a disparidade de RY e MK (b), a disparidade AT e GC (c) e curva da disparidade de AT ao longo do cromossoma (d).

2.3.2 Representações numéricas

Para além da motivação em obter representações gráficas, as representações numéricas também permitem uma análise das sequências de ADN através de metodologias tradicionalmente usadas em dados numéricos. De seguida serão apresentadas algumas dessas representações.

Um dos esquemas de mapeamento mais populares é a representação binária de *Voss* [38]. Esta representação consiste em mapear os quatro nucleótidos (A, C, G e T) em sequências indicadoras binárias, que indicam a presença (1) e ausência (0) dos respectivos nucleótidos. O indicador binário define-se então por:

$$u_k[x_n] = \begin{cases} 1 & \text{se } x_n = k, \\ 0 & \text{se } x_n \neq k, \end{cases}$$

onde $k \in \mathcal{A} = \{A, C, G, T\}$ e n representa o índice da base ($n = 0, 1, 2, ..., N-1$). Garantindo sempre que $u_A[x_n] + u_C[x_n] + u_G[x_n] + u_T[x_n] = 1$.

Assim, para a sequência $x = AGTTCTACCGAGC$ as sequências indicadoras binárias para cada base seriam:

$$u_A[x] = \{1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0\}$$
$$u_C[x] = \{0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1\}$$
$$u_G[x] = \{0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0\}$$
$$u_T[x] = \{0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0\}$$

Esta representação foi adaptada por *Anastassiou* [2], através da atribuição de números complexos (a, t, c, g) às sequências indicadoras. Desta forma, a sequência numérica resultante de um fragmento de letras de tamanho N é escrita como:

$$x[n] = au_A[n] + tu_T[n] + cu_C[n] + gu_G[n], \qquad n = 0, 1, 2, ..., N-1$$

onde u_A, u_T, u_C e u_G são as sequências indicadoras binárias que, tal como na representação de *Voss*, tomam o valor 1 ou 0, dependendo se a letra correspondente existe ou não na posição n.

A escolha apropriada das constantes a, t, c e g pode evidenciar propriedades úteis da sequência numérica x. Por exemplo, se forem escolhidas as constantes no conjunto dos complexos de tal forma que $t = a^*$ e $g = c^*$, com b^* o conjugado complexo de b, tem-se que a cadeia complementar é dada por y, com:

$$y[n] = x^*[-n + N - 1], \qquad n = 0, 1, ..., N-1$$

Uma das muitas possíveis atribuições é a seguinte:

$$a = 1 + j, \quad t = 1 - j, \quad c = -1 - j, \quad g = -1 + j$$

Assim, utilizando a mesma sequência que no exemplo anterior ($CGAT$), obtém-se a seguinte sequência:

$$x = [-1 - j, -1 + j, 1 + j, 1 - j]$$

No entanto, é possível reduzir o número de sequências indicadoras de quatro para três, através do método do tetraedro [34]. Neste método, cada uma das quatro letras é mapeada num vértice de um tetraedro regular no espaço. Assim, três sequências numéricas (x_r, x_g e x_b) são definidas a partir dos coeficientes:

$$(a_r, t_r, c_r, g_r), \quad (a_g, t_g, c_g, g_g), \quad (a_b, t_b, c_b, g_b)$$

E considera-se que os quatro vectores 3D têm magnitude 1 e apontam para as quatro direcções do centro para os vértices do tetraedro. Pode-se então escolher os seguintes valores, por exemplo:

$$\begin{aligned}
(a_r, a_g, a_b) &= (0, 0, 1) \\
(t_r, t_g, t_b) &= (\tfrac{2\sqrt{2}}{3}, 0, \tfrac{-1}{3}) \\
(c_r, c_g, c_b) &= (-\tfrac{\sqrt{2}}{3}, \tfrac{\sqrt{6}}{3}, \tfrac{-1}{3}) \\
(g_r, g_g, g_b) &= (-\tfrac{\sqrt{2}}{3}, -\tfrac{\sqrt{6}}{3}, \tfrac{-1}{3})
\end{aligned}$$

Obtendo-se então as três sequências indicadoras seguintes:

$$\begin{aligned}
x_r[n] &= \tfrac{\sqrt{2}}{3}(2u_T[n] - u_C[n] - u_G[n]) \\
x_g[n] &= \tfrac{\sqrt{6}}{3}(u_C[n] - u_G[n]) \\
x_b[n] &= \tfrac{1}{3}(3u_A[n] - u_T[n] - u_C[n] - u_G[n])
\end{aligned}$$

A partir das sequências indicadores de qualquer um deste métodos descritos, têm sido calculadas as transformadas discretas de *Fourier* que permitem obter informação sobre a frequência das quatro bases das sequências de ADN.

2.3.3 Distâncias inter-simbólicas

Nair e Mahalakshmi [24] introduziram a distância inter-nucleótidos, uma representação numérica que permite explorar a correlação da estrutura do ADN, estendida mais tarde por Afreixo *et al.* [1]. Esta representação consiste em converter sequências de ADN em sequências numéricas do mesmo tamanho, onde cada número representa a distância que separa o símbolo actual da próxima ocorrência do mesmo símbolo. Por exemplo, considerando a sequência circular $AAACCCGTGTCAGTT$, a sequência global da distância inter-nucleótidos seria:

$$d = (1, 1, 9, 1, 1, 5, 2, 2, 4, 4, 8, 4, 9, 1, 8)$$

Da mesma forma, podem-se obter quatro sequências de distâncias inter-nucleótidos, associadas a cada um dos nucleótidos:

$$d^A = (1, 1, 9, 4), \quad d^C = (1, 1, 5, 8), \quad d^G = (2, 4, 9), \quad d^T = (2, 4, 1, 8)$$

Como observado em [1], o comprimento da sequência global de distâncias inter-nucleótidos d é igual à soma dos comprimentos das quatro sequências de distâncias inter-nucleótidos:

$$N = N_A + N_C + N_G + N_T$$

Desta forma, quando a posição da primeira ocorrência de cada nucleótido é fornecida (k_0^A, k_0^C, k_0^G, k_0^T), é possível determinar a posição de todos os nucleótidos da sequência completa através da sequência de distâncias inter-nucleótidos:

$$k_j^x = \sum_{i=1}^{j} d_i^x + k_0^x$$

onde d_i^x é a distância que se encontra no índice i da sequência de distâncias inter-nucleótidos do nucleótido x.

Com o objectivo de estudar algumas propriedades estatísticas, foi usada a distribuição de referência da probabilidade de distâncias inter-nucleótidos. Os autores consideraram como distribuição de referência a distribuição geométrica onde se assume que as sequências de nucleótidos foram geradas por um processo aleatório independente e igualmente distribuído. A função de distribuição de referência é definida por:

$$F^x(k) = P(d^x \leqslant k) = 1 - (1 - p^x)^k$$

com um valor esperado de $1/p^x$ e uma variância de $(1 - p^x)/(p^x)^2$. Para estimar a probabilidade de ocorrência, p^x, dos quatro nucleótidos foram calculadas as frequências relativas desses nucleótidos na sequência de ADN. Na Figura 2.11a são apresentas as distribuições das frequências relativas, resultantes deste estudo.

Foi ainda feita uma análise do erro relativo para a distribuição das distâncias inter-nucleótidos do genoma humano completo, verificando-se que as duas primeiras distâncias têm uma maior frequência que as sequências aleatórias correspondentes e as próximas dez distâncias têm menores frequências (Figura 2.11b). Para a distribuição das distâncias inter-nucleótidos da região codificante as primeiras distâncias continuam a ter uma frequência mais alta do que as sequências aleatórias correspondentes, mas o erro relativo mostra um comportamento oscilatório para as primeiras distâncias que poderá indicar alguma periodicidade subjacente.

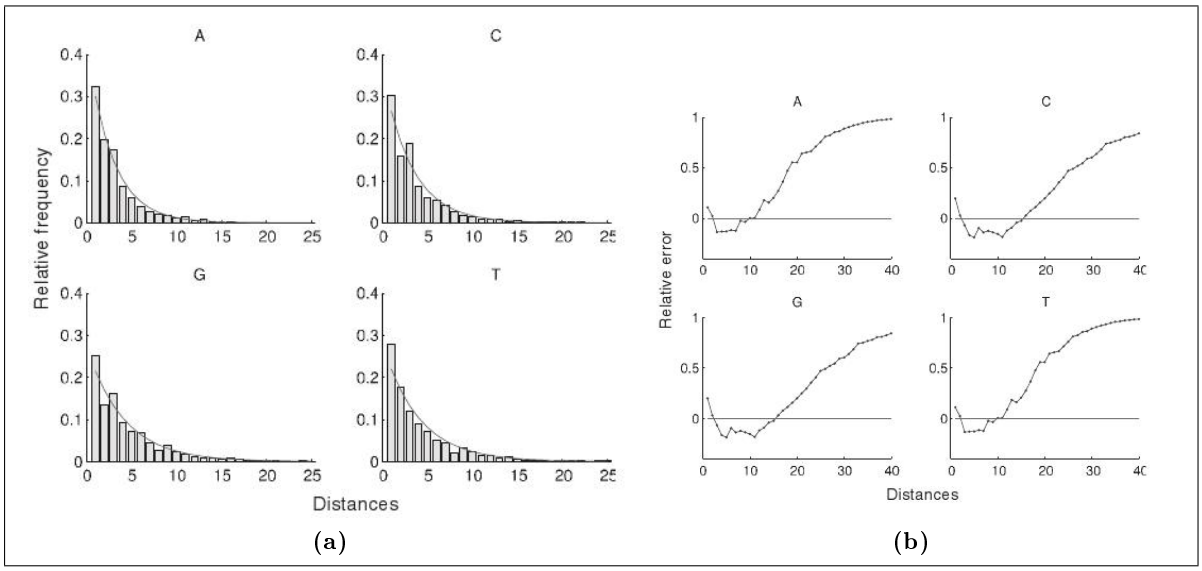


Figura 2.11: Distribuição da frequência relativa das quatro sequências de distâncias inter-nucleótidos do gene $gi|33286443|ref|NM\_032427.1$ do *Homo sapiens* (a) e erro relativo das distâncias nucleotídicas do genoma completo do *Homo sapiens* (b). (Figura retirada de [1])

Este estudo sugere que o mapeamento das distâncias inter-simbólicas caracteriza uma sequência de ADN e também que existe uma assinatura genética para cada espécie.

O mesmo estudo foi aplicado a distâncias inter-simbólicas de dois nucleótidos [4]. De maneira a comparar as distribuições das frequências relativas de cada dinucleótido, foi calculada a divergência de *Kullback-Leibler* entre as distribuições empíricas dos 16 dinucleótidos do genoma humano (Tabela 2.1). Como se pode verificar, os resultados da divergência entre os dinucleótidos e o seu complemento invertido têm valores mais pequenos, o que significa que têm um maior grau de semelhança em relação aos outros pares.

Tabela 2.1: Tabela da divergência *Kullback-Leibler* entre os 16 dinucleótidos do genoma humano. (Tabela retirada de [4])

	AA	**AC**	**AG**	**AT**	**CA**	**CC**	**CG**	**CT**	**GA**	**GC**	**GG**	**GT**	**TA**	**TC**	**TG**	**TT**
AA	0.000	0.275	0.106	0.061	0.112	0.211	1.190	0.105	0.147	0.610	0.211	0.273	0.090	0.147	0.111	0.000
AC	0.316	0.000	0.086	0.143	0.110	0.041	0.649	0.086	0.025	0.135	0.041	0.000	0.074	0.025	0.111	0.318
AG	0.105	0.076	0.000	0.015	0.006	0.052	0.875	0.000	0.025	0.220	0.052	0.075	0.013	0.025	0.006	0.106
AT	0.058	0.125	0.014	0.000	0.013	0.078	0.950	0.014	0.048	0.288	0.078	0.124	0.016	0.048	0.013	0.059
CA	0.106	0.090	0.005	0.013	0.000	0.069	0.905	0.005	0.036	0.228	0.069	0.089	0.020	0.036	0.000	0.107
CC	0.240	0.044	0.074	0.099	0.104	0.000	0.657	0.074	0.032	0.104	0.000	0.044	0.040	0.032	0.105	0.242
CG	2.966	2.144	2.894	2.755	3.291	1.682	0.000	2.891	2.312	1.565	1.692	2.158	2.043	2.313	3.311	2.969
CT	0.105	0.076	0.000	0.015	0.006	0.052	0.875	0.000	0.025	0.220	0.052	0.076	0.013	0.025	0.006	0.106
GA	0.167	0.025	0.028	0.054	0.043	0.030	0.753	0.028	0.000	0.194	0.030	0.024	0.019	0.000	0.043	0.169
GC	0.521	0.093	0.221	0.281	0.258	0.079	0.571	0.222	0.137	0.000	0.079	0.094	0.180	0.137	0.260	0.525
GG	0.239	0.044	0.074	0.099	0.104	0.000	0.657	0.074	0.031	0.104	0.000	0.044	0.040	0.032	0.105	0.242
GT	0.314	0.000	0.085	0.142	0.109	0.041	0.651	0.085	0.024	0.135	0.041	0.000	0.074	0.024	0.110	0.316
TA	0.097	0.072	0.016	0.019	0.028	0.038	0.830	0.016	0.019	0.228	0.038	0.071	0.000	0.019	0.028	0.098
TC	0.167	0.025	0.028	0.054	0.043	0.030	0.754	0.028	0.000	0.194	0.030	0.024	0.019	0.000	0.043	0.169
TG	0.105	0.091	0.005	0.013	0.000	0.069	0.907	0.005	0.036	0.229	0.069	0.090	0.020	0.036	0.000	0.107
TT	0.000	0.277	0.107	0.062	0.113	0.212	1.192	0.106	0.149	0.613	0.212	0.275	0.091	0.149	0.112	0.000

O estudo das distâncias inter-simbólicas revelou ser útil para caracterizar uma sequência de ADN, tanto para símbolos de uma letra (nucleótidos) como de duas (dinucleótidos), o que leva a crer que nas distâncias entre símbolos com mais de duas letras também se continuem a verificar resultados interessantes que possam contribuir para o estudo da evolução das espécies.

2.4 Exemplos de ferramentas existentes

Existem diversas ferramentas de análise de sequências de ADN que visam contribuir para um estudo mais aprofundado da sua estrutura e do seu funcionamento. Estas ferramentas dividem-se em diferentes categorias consoante o tipo de análise que se pretende fazer. Algumas dedicam-se ao estudo da estrutura primária, outras ao alinhamento de sequências, outras ainda à procura de sequências, redesenho de genes ou até identificação de Ácido Ribonucleico de Transferência (ARNt).

De facto, a oferta de ferramentas nesta área é muito vasta tornando impossível a descrição de todas e obrigando a uma selecção criteriosa das mesmas. Assim, nesta secção iremos abordar cinco ferramentas de uso livre e de diferentes categorias, por forma a mostrar o panorama geral das aplicações existentes e fazer uma pequena descrição das suas funcionalidades.

2.4.1 *ANACONDA*

ANACONDA [23] foi desenvolvido no âmbito de uma dissertação de doutoramento da Universidade de Aveiro em 2010. Baseia-se essencialmente no estudo da estrutura primária dos genes utilizando um conjunto de métodos estatísticos e de visualização de maneira a fornecer uma análise do contexto de codões e identificar zonas onde existem repetições dos mesmos.

O conjunto de dados genéticos pode ser carregado a partir de bases de dados públicas para posterior análise, utilizando diferentes métodos estatísticos como *clustering, biclustering*, análise residual e índices de adaptação dos codões. Assim, a ferramenta lê todos os genes presentes num determinado genoma e constrói tabelas de contingência (Figura 2.12).

ANACONDA é uma aplicação que permite efectuar caracterizações de genes, obtendo vários índices, podendo também efectuar comparações entre ambos. É possível também redesenhar genes, com base nos vários valores estatísticos obtidos através da aplicação. Esta ferramenta foi desenvolvida em C++ e só corre em *Windows*.

2.4.2 *REPuter*

REPuter [8, 17] centra-se no estudo sistemático da estrutura repetitiva de genomas completos. Fornece uma descrição completa das repetições directas e palindrómicas, assim como a detecção de repetições complementares e invertidas, como nos exemplos seguintes:

Repetições directas: $GAGTC \Longleftrightarrow GAGTC$
Repetições palindrómicas: $GAGTC \Longleftrightarrow CTGAG$
Repetições complementares: $GAGTC \Longleftrightarrow CTCAG$
Repetições invertidas: $GAGTC \Longleftrightarrow GACTC$

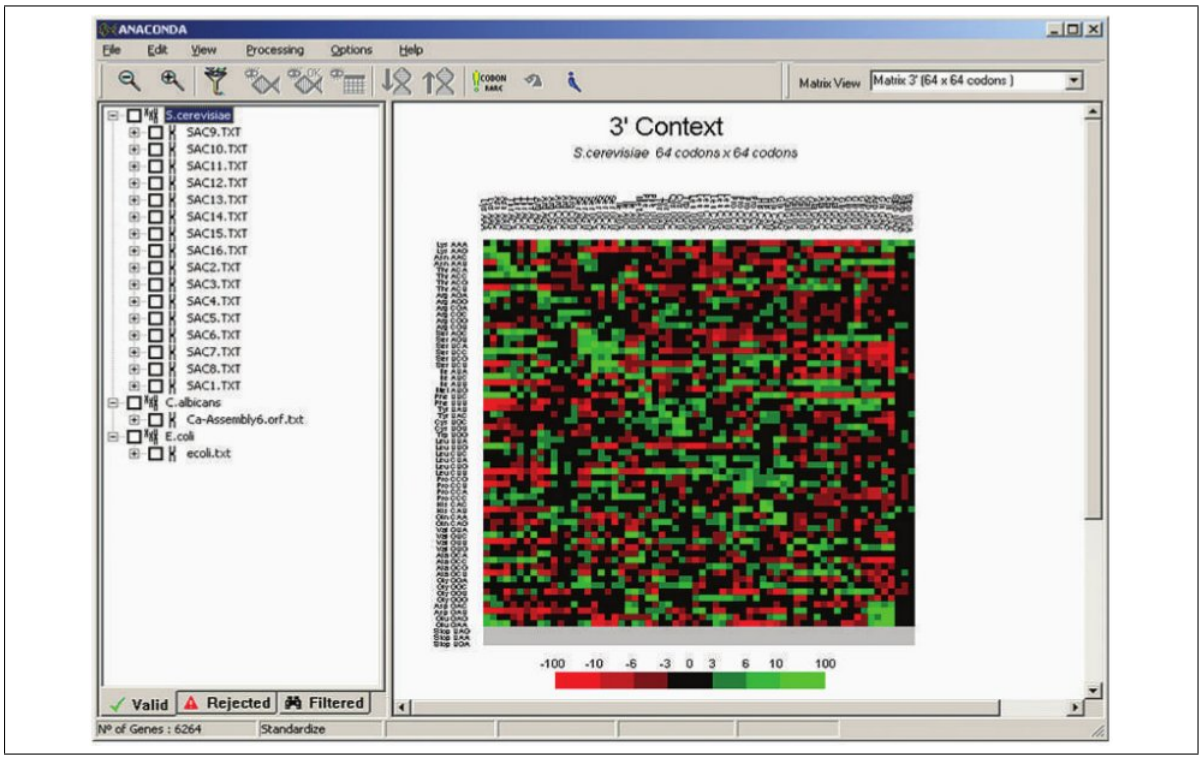


Figura 2.12: Janela principal da ferramenta *ANACONDA* a apresentar o mapa genómico da análise do contexto de codões do *S. cerevisiae*.

REPuter é uma família de ferramentas, sendo uma delas o *REPfind* que consiste numa implementação para localizar repetições em tempo linear, gerando então um ficheiro volumoso que pode ser visualizado pelo *REPvis* (Figura 2.13). Outra ferramenta incluída é o *REPselect* que permite seleccionar zonas de repetições que revelem maior importância para que possam ser posteriormente guardadas.

O *REPuter* proporciona assim a possibilidade de comparar duas ou mais sequências utilizando diferentes níveis de semelhança, que podem ser controlados através da alteração das opções que determinam a taxa de erro a utilizar. Esta ferramenta encontra-se disponível para os sistemas operativos *Linux, OSX e Solaris*.

2.4.3 *GraphDNA*

A ferramenta *GraphDNA* [37] foi desenvolvida por *Chris Upton* do *Viral Bioinformatics Resource Center* da Universidade de Victoria no Canadá e permite gerar representações gráficas de caminhos (Figura 2.14) a partir de sequências de ADN.

Estes caminhos são criados utilizando diferentes representações seleccionadas previamente pelo utilizador, existindo oito opções diferentes de processamento. A ferramenta suporta análise de genes individuais assim como de genomas completos.

Foi desenvolvida na linguagem Java, podendo ser usada nos sistemas operativos *Mac OS X, Windows e Linux*.

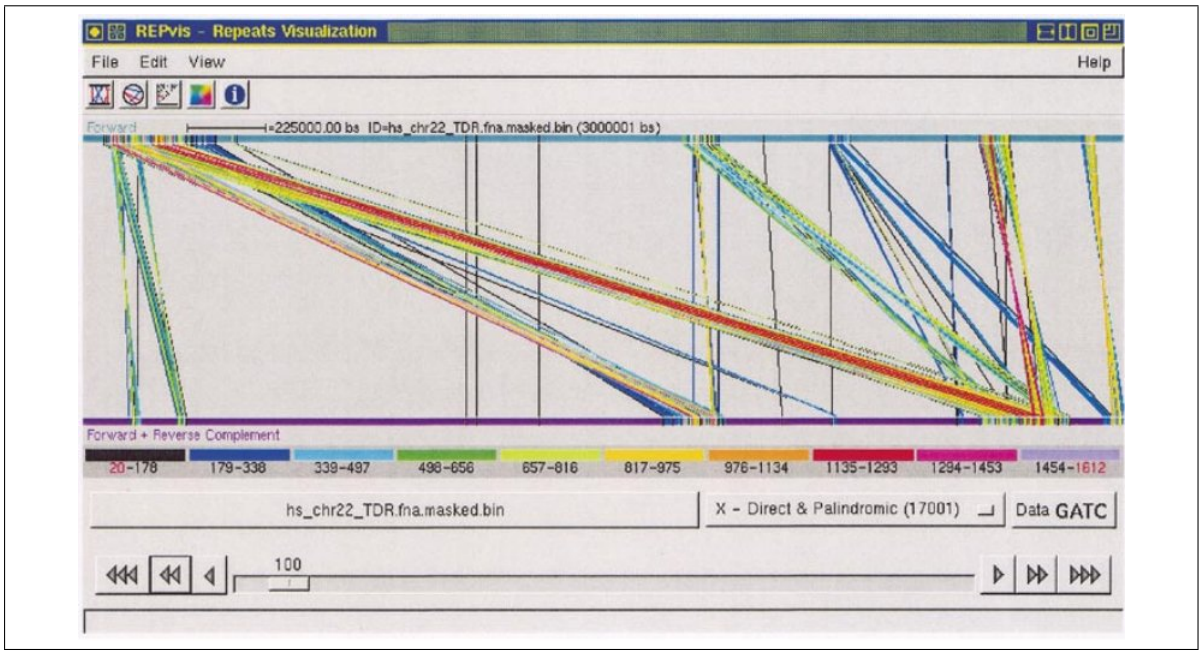


Figura 2.13: Ferramenta REPuter: exemplo da janela REPVis com gráfico resultante das repetições directas e palindrómicas do cromossoma humano 22 com o tamanho mínimo de 100bp

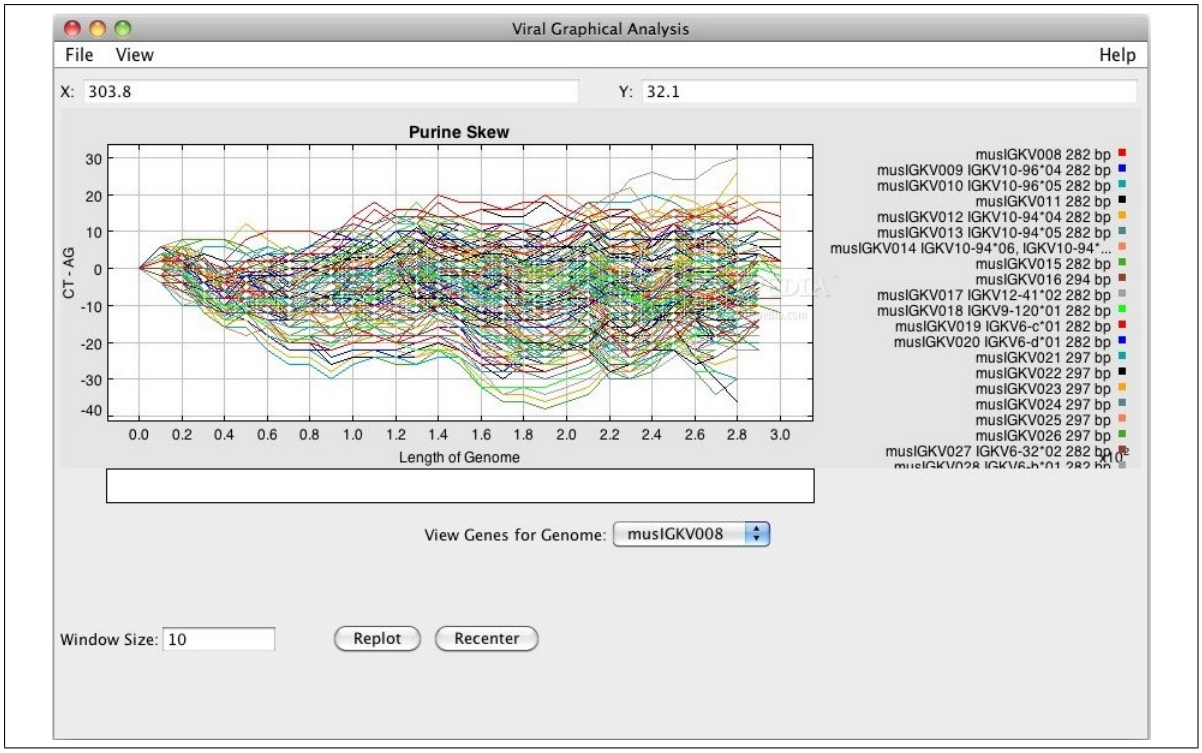


Figura 2.14: Janela principal da ferramenta *GraphDNA* com a representação gráfica de vários genomas em simultâneo

2.4.4 *BioEdit*

O *BioEdit* [13] é uma ferramenta orientada para a edição, manipulação e análise de sequências. Foi criada em 1999 por um estudante universitário com o objectivo de melhorar o trabalho exaustivo da utilização de programas biológicos baseados em linhas de comandos.

Esta ferramenta permite fazer as mais simples acções de edições e de alinhamento múltiplo através do uso de uma interface gráfica intuitiva (Figura 2.15). Oferece ainda ao utilizador a possibilidade de fixar certas posições nas sequências, efectuando posteriormente o alinhamento sem que essas posições sejam alteradas. Assim sendo, permite o congelamento de certos domínios conhecidos, alinhando as restantes partes em torno desses domínios. Esta ferramenta apenas corre em *Windows*.

2.4.5 *GeneXpress*

GeneXPress [33] é uma ferramenta de visualização e análise de expressão genética. Foi desenvolvida para facilitar a atribuição de significados biológicos a determinados padrões de expressão genética através de processos automáticos.

Assim, é possível identificar os processos biológicos representados por cada padrão (Figura 2.16) e obter múltiplas visualizações dos perfis de expressão, incluindo vistas globais e vistas detalhadas. Através de análises estatísticas relativas a bases de dados de anotações de genes, esta ferramenta pode associar cada padrão a um ou mais processos biológicos.

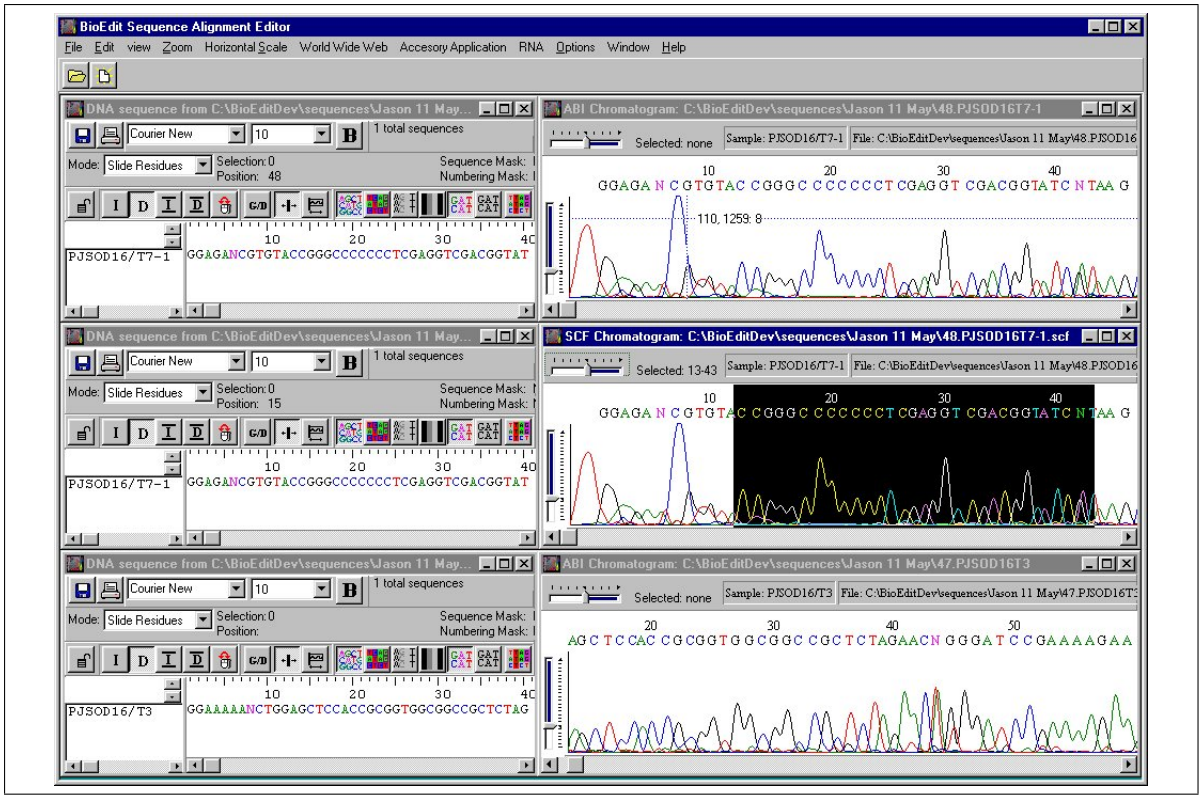


Figura 2.15: Exemplo de uma das janelas da ferramenta *BioEdit* que mostra a vista de edição, conversão e alinhamento de uma sequência.

Figura 2.16: Janela principal da ferramenta *GeneXPress* que apresenta um gráfico para cada padrão genético, fornecendo informação sobre a percentagem de genes que contem esse padrão.

2.5 Repositório de dados

Existe uma grande quantidade de informação genética que se encontra dispersa por diversas bases de dados e em distintos formatos.

No artigo publicado anualmente pela revista *Nucleic Acids Research* (NAR) podemos obter informação actualizada sobre as bases de dados de Biologia Molecular disponíveis na *web*, que conta actualmente com a existência de 1330 bases de dados distintas [10]. Estas bases de dados podem ser generalistas ou focarem-se em estudos mais específicos. Existindo assim umas que se dedicam à informação de sequências de nucleótidos ou de sequências de proteínas de várias espécies, outras que contêm exclusivamente informação sobre o genoma humano e doenças, reacções enzimáticas, etc.

A informação divide-se ainda em dois tipos de dados, aqueles que são suportados por sistemas de ficheiros e os suportados por bases de dados relacionais. Devido à dimensão que normalmente este tipo de dados possui, é vantajoso utilizar informação disponível através de ficheiros de maneira a tornar o seu processamento mais eficiente.

Neste trabalho, iremos então focar-nos nas bases de dados de sequências de nucleótidos suportadas por sistemas de ficheiros. As bases de dados principais a nível mundial são a *European Molecular Biology Laboratory* (EMBL), a *DNA Data Bank of Japan* (DDBJ) e o *Genetic Sequence Databank* (GenBank). Com o objectivo de proporcionar uma colecção global de sequências genéticas com um domínio público, foi criado um protocolo de cooperação, o *International Nucleotide Sequence Database Collaboration* (INSDC), que permite a troca diária de informação entre as três principais bases de dados de uma forma eficiente e organizada.

Qualquer uma destas bases de dados pode ser acedida através de ligações FTP, permitindo descarregar facilmente todos os ficheiros de sequências genéticas que se encontrem disponíveis.

2.6 Formatos de dados

Existem diferentes formatos de dados usados para representar as sequências de nucleótidos, que variam consoante o tipo de informação que se pretende fornecer ou a fonte da mesma, sendo os formatos mais comuns o GenBank, EMBL e FASTA.

O GenBank descreve as sequências de uma forma muito completa, fornecendo o nome do organismo, a taxonomia, mutações e repetições, identificação das zonas de codificação, entre outros dados. Este ficheiro é composto por um conjunto de marcadores (Figura 2.17), começando a primeira linha com o marcador LOCUS que contém o nome e o número de bases da sequência. O início da sequência é marcado pela palavra ORIGIN e termina com "//". Existem diversos marcadores que podem surgir, como o DEFINITION que contém o nome do organismo a que pertence a sequência ou o FEATURES que contém as posições dos genes na sequência.

```
LOCUS        scaffold129153 1185 bp DNA HTG 2-SEP-2011
DEFINITION   Felis catus scaffold scaffold129153 CAT full sequence 1..1185 reannotated via EnsEMBL
KEYWORDS     .
SOURCE       domestic cat
ORGANISM     Felis catus
             Eukaryota; Metazoa; Eumetazoa; Bilateria; Coelomata; Deuterostomia;
FEATURES     Location/Qualifiers
             source 1..1185
             organism="Felis catus"
             dbxref="taxon:9685"
             miscfeature 1..1185
             note="contig contig29859 1..1185(1)"
BASE COUNT   215 a 321 c 378 g 271 t
ORIGIN
               1 AGAGCTCCCC CAATACCCCC TTCCGCAAGG TATGGGGAGC CCCCCTGCAC TGGGGCAGGG
              61 CGCAGGGAAG GTTGGGGGCT GGGGTTGAAG ACTTGGTCTG AGTCTACTGG AGGGATGGCT
             121 ATGGGGTGAT GCAGAGGGGC TGATGTACAG ATTTCGTCTG TCAACCCCAG GACCTCATCA
             181 GCCTGGACAC GTCCCCGGCT AAGGAGCGGC TGGAGGAGAG TTGTGTGCAC CCCCTCGAAG
```

Figura 2.17: Exemplo de sequência no formato GenBank.

O formato EMBL é estruturado por linhas (Figura 2.18) onde cada uma tem um código composto por dois caracteres que indica o tipo de informação que se segue. O ficheiro começa sempre com o identificador "ID" e termina com a linha que contém "//". O início da sequência está identificado pelo marcador "SQ".

Num ficheiro FASTA, a primeira linha começa com o símbolo ">" e contém a identificação da sequência, que é composta pelo nome da sequência e a sua descrição, normalmente separados com o símbolo "|" (Figura 2.19). Nas linhas seguintes encontra-se a sequência de nucleótidos. Habitualmente os ficheiros têm 60 caracteres por linha, sendo aconselhável não ultrapassar os 80. O aparecimento do símbolo ">" indica o início de uma nova sequência, uma vez que cada ficheiro pode conter uma ou mais sequências. A extensão genérica deste formato é *.fasta* mas existem outras extensões tais como *.fa*, *.fsa*, *.fna*, *.ffn*, *.faa* e *.frn*.

```
ID   scaffold211934 standard; DNA; HTG; 851 BP.
DT   2-SEP-2011
XX
DE   Felis catus scaffold scaffold211934 CAT full sequence 1..851 annotated by Ensembl
KW   .
OS   Felis catus (domestic cat)
OC   Eukaryota; Metazoa; Eumetazoa; Bilateria; Coelomata; Deuterostomia;
XX
FH   Key Location/Qualifiers
FT   source 1..851
FT   /organism="Felis catus"
FT   /db-xref="taxon:9685"
FT   misc-feature 1..851
FT   /note="contig contig-629815 1..851(1)"
SQ   Sequence 851 BP; 326 A; 159 C; 149 G; 217 T; 0 other;
     CAGAGCACAT TCCGTATTAT ATTGGAATTA TTTTTTTACC ATCTATTTTT CCCAATTAAG       60
     TGCAAACTCT TCAAAGAGAG GATCCATGTT TTATTAATTT AGTTTATTCT AGTGTTCTGA      120
     TATACAGTAG GTAGTCAGTA AATGTTTATT TACTTTATTG AAATTTGATG ATTAATATAT      180
     TGTATATACC AAGTGTATTT TGGGGGCTGG CACACATCCA ATGTCTTCAT GTTTTTCCAG      240
```

Figura 2.18: Exemplo de sequência no formato EMBL.

```
>ENSFCAT00000005424 cdna:novel scaffold:CAT:scaffold-185945:
37728:43812:-1 gene:ENSFCAG00000005422
ATGGCTGGAGAATCGGCGGAGAGCACAGCCTGGGACGCCCAGCCCACAGAGAACCACATG
GGGCCTCTGGTGGCGAAGACGGAAGCCGGAGAGGCCTCTGCCCCCGTGCGGGACGCCAGC
CCCCATCGGGGTCCTGAACACTCGCGCCGGCGCTTCCGGGGCTTCCGCTACCCTGAGGCC
GAGGGGCCCCGCGAGGCGCTGAGCCGGCTCCGCGAGCTGTGCCGCCAGTGGCTGAGCCCG
GATATACACACCAAGGAGCAGATCCTGGAGCTGCTGGTGCTGGAGCAGTTCCTGACCATC
CTGCCCGCCGAGCTCCAGGCCTGGGTGCGGGGACAGCACCCGGAGAGCGGGGACGAGGTG
GTGGTGCTCCTGGAGCACCTGCAGAGACAGCTGGAGGCGCCGACACCGCAGGTCCCAGGT
GGTGACCAGGGCCAAGAGCTTGTCTGTCGCGAGATGGCAGCACTGGCACCTTCCCGCGGA
TCACGGAGTGCCCAGTTCCAGCCGGTGAGGGCTCTGCTCAAGCATGAATCTCTGGGATCA
CGGGCCTTACCGGGCACAGTTCTCCAGGGTCCTGGGCTTGCCCCGGGAGGGCGCTGCAGA
GGAGACGCAGTGGTGGCGGCCAGGCTGCCGCCAGAGCCCCAGGGCTTGCTGAAAACGGAA
```

Figura 2.19: Exemplo de sequência no formato FASTA.

Capítulo 3

AGenDA

A grande quantidade de dados genéticos existente fez com que surgissem variados métodos de análise do ADN e consequentemente a criação de inúmeras ferramentas computacionais, como as exemplificadas no capítulo anterior.

AGenDA é o acrónimo de *A Genome Distance Analyser* e foi o nome dado à ferramenta desenvolvida neste trabalho, que tem como objectivo servir de suporte à análise do mapeamento de distâncias inter-simbólicas do ADN.

Este capítulo engloba todas as etapas do desenvolvimento da ferramenta AGenDA, que incluem a análise de requisitos, arquitectura do sistema, a implementação da ferramenta e a sua documentação.

3.1 Análise de requisitos

Definir os requisitos da ferramenta constitui uma etapa fundamental no desenvolvimento da mesma. Nesta secção serão primeiro enumerados os requisitos funcionais, que definem aquilo que a ferramenta deverá fazer, e de seguida os requisitos não funcionais, que definem o comportamento que a ferramenta deverá ter. No final será apresentado um diagrama que mostra de uma forma mais perceptível as funcionalidades que se pretende que a ferramenta tenha.

3.1.1 Requisitos funcionais

Os requisitos funcionais definem aquilo que o sistema deve fazer e são normalmente determinados através da interacção directa com os futuros utilizadores.

Neste trabalho, não foram utilizadas as técnicas habituais de entrevistas e questionários dirigidas aos utilizadores com o objectivo de saber aquilo que se pretende que a ferramenta faça. Os principais requisitos já se encontravam definidos na proposta inicial do projecto e ao longo do seu desenvolvimento foram surgindo novas ideias de funcionalidades que foram posteriormente discutidas. Depois de analisadas e seleccionadas as diferentes sugestões propostas, definiram-se os seguintes requisitos funcionais:

- suportar ficheiros de dados de sequências de nucleótidos no formato FASTA;
- processar as distâncias inter-nucleótidos dos ficheiros de dados;
- fazer análises estatísticas baseadas nas distâncias inter-nucleótidos;

- definir diferentes parâmetros de processamento: tamanho da palavra, distância máxima a considerar, tipos de análise a fazer;
- seleccionar áreas locais de análise;
- permitir visualizar resultados estatísticos de uma forma gráfica;
- oferecer a possibilidade de gravar gráficos e tabelas de resultados em ficheiros no disco;

3.1.2 Requisitos não funcionais

Os requisitos não funcionais, também conhecidos por atributos do sistema, expressam o comportamento do mesmo nomeadamente no que diz respeito a questões de desempenho e de usabilidade.

Como já foi dito, os ficheiros de sequências genéticas são normalmente ficheiros de grandes dimensões, sendo o tempo de resposta e os requisitos mínimos de memoria duas características fundamentais a ter em conta no desenvolvimento desta ferramenta. Para além disso, é importante criar uma aplicação com uma interface gráfica intuitiva e que funcione em diferentes plataformas, permitindo assim que os futuros utilizadores a possam utilizar em qualquer sistema operativo.

3.1.3 Casos de uso

Com o objectivo de ilustrar de uma forma mais detalhada os requisitos da aplicação, utilizou-se um diagrama de casos de uso, baseado na notação gráfica *Unified Modeling Language* (UML) [3]. Este modelo permite visualizar de uma forma mais concreta as funcionalidades do sistema através da definição dos casos de uso, da identificação dos actores e das relações existentes entre eles. Os casos de uso definem então o comportamento do sistema, sendo uma maneira diferente e complementar de enumerar os requisitos funcionais. Os actores são entidades externas que interagem com o sistema e executam determinadas tarefas, podendo ser pessoas, como por exemplo os utilizadores, ou outros sistemas.

A Figura 3.1 representa o diagrama de casos de uso da ferramenta, onde se pode verificar a existência de um único actor, o utilizador. As duas funcionalidades base consistem no processamento das distâncias a partir de um ficheiro no formato FASTA e na criação de gráficos e tabelas de resultados a partir de um ficheiro binário.

O processamento de distâncias divide-se em distâncias globais e distâncias locais. Para qualquer um destes processamentos, o utilizador deverá carregar o ficheiro ou os ficheiros que pretende analisar, podendo eliminar ficheiros que já tenham sido carregados para a área pessoal. Para além disso, poderá alterar as opções de processamento, seleccionando aquelas que pretende efectuar (análise conjunta, análise individual, regiões codificantes ou *reading frame*), definindo os valores da distância máxima a considerar e do tamanho da palavra.

Se o utilizador optar pelo processamento das distâncias globais, poderá escolher se serão processados todos os ficheiros presentes na área pessoal ou se apenas um ficheiro seleccionado. Caso opte pelas distâncias locais, deverá ainda especificar qual a área que pretende analisar. Esta especificação pode ser feita de um modo visual, seleccionando a área pretendida com o rato, ou então através de parâmetros para delimitar a área.

Uma vez feito o processamento das distâncias globais ou locais, o utilizador terá a possibilidade de guardar os dados processados. Os dados poderão ser armazenados em ficheiros

separados que contenham tabelas e imagens com gráficos, ou então num único ficheiro binário onde será guardada toda a informação do processamento para posterior visualização de resultados.

A segunda funcionalidade consiste então em ler o ficheiro binário que foi previamente criado pelo programa. O utilizador poderá consultar toda a informação do ficheiro, tais como os nomes dos ficheiros FASTA processados e os parâmetros usados para calcular os resultados, e utilizar esses dados para construir tabelas e gráficos estatísticos. Neste caso, os resultados serão mais rápidos uma vez que não se efectua o processamento completo das distâncias de todos os ficheiros, apenas se processam os resultados finais.

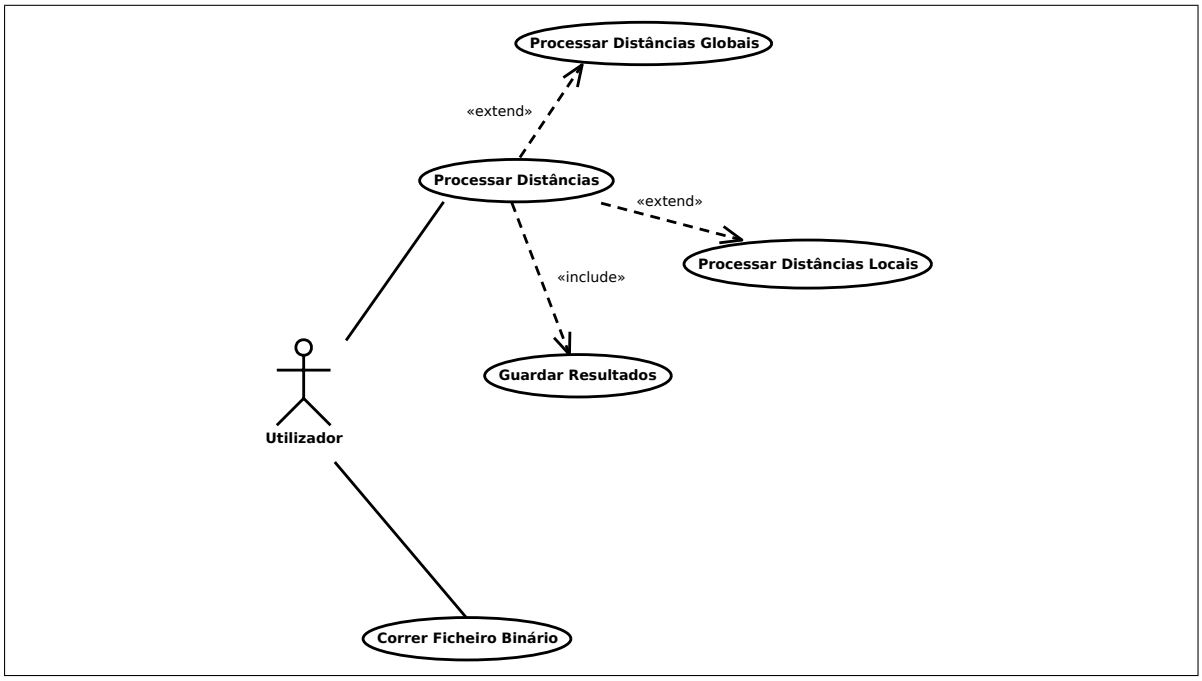


Figura 3.1: Diagrama de casos de uso da ferramenta AGenDA.

3.2 Arquitectura

Uma vez definidas as funcionalidades da ferramenta, ou seja, aquilo que a ferramenta deve fazer, é necessário estruturar como as iremos pôr em prática.

Neste capítulo serão então definidas as estruturas das classes utilizadas para a programação da ferramenta e será feita uma descrição das mesmas. Para além disso, serão também apresentadas as estruturas de dados usadas para armazenar a informação necessária ao processamento das distâncias inter-simbólicas.

3.2.1 Diagrama de classes

Nesta fase de desenvolvimento, optou-se por usar um diagrama de classes UML [3], que fornece uma visão gráfica dos principais elementos arquitectónicos do sistema através da definição da estrutura das classes que o constituem. Uma vez determinadas quais as classes existentes é necessário determinar quais os seus atributos, as suas operações e as relações existentes com as outras classes. Para já será apenas apresentado um diagrama de classes muito

geral, que apenas mostra as diferentes classes existentes e as suas relações, sendo feita uma descrição mais detalhada de algumas classes na secção seguinte.

Como podemos observar através da Figura 3.2, o sistema encontra-se dividido em três módulos fundamentais: interface gráfica (*gui*), dados (*data*) e processamento (*processing*).

O primeiro módulo engloba tudo o que diz respeito a janelas gráficas e métodos de interacção com o utilizador. É constituído pela classe principal do programa, a *MainWindow*, que se divide em cinco classes que definem os diferentes tipos de janelas existentes:

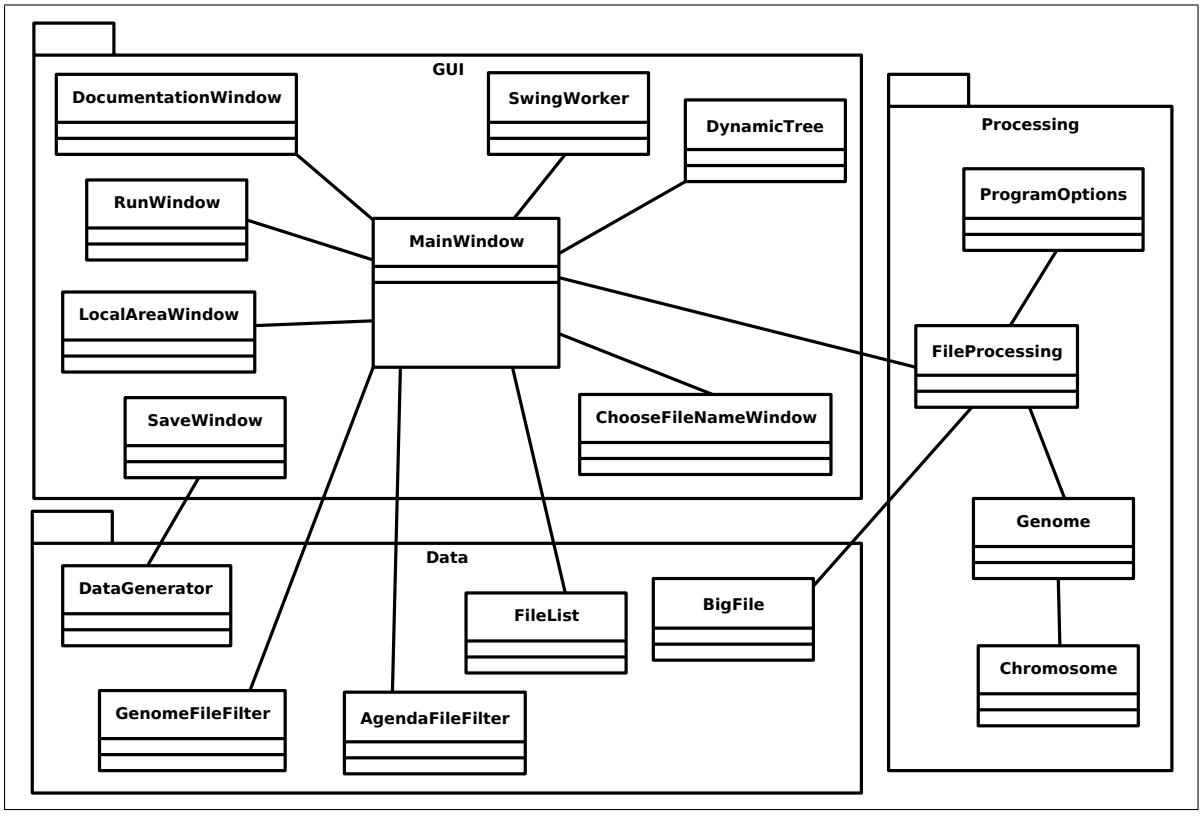


Figura 3.2: Diagrama de classes da ferramenta AGenDA.

- *DocumentationWindow*: para abrir a documentação da ferramenta;
- *RunWindow* para informar o estado do processamento de dados;
- *LocalAreaWindow*: para seleccionar a área local a analisar;
- *SaveWindow*: para mostrar as diferentes opções de armazenamento;
- *ChooseFileNameWindow*: para seleccionar o ficheiro ou directório que se pretende abrir.

Possui também a classe *SwingWorker*, uma classe abstracta utilizada em aplicações *Swing* para realizar tarefas mais longas num *thread* dedicado. Este tema será aprofundado na secção 3.3.4. Contém ainda a classe *DynamicTree*, que consiste numa implementação que organiza informação dos diferentes directórios em árvore, permitindo que o utilizador possa navegar ao longo da mesma expandindo os seus ramos.

O módulo *processing* é constituído pela classe *FileProcessing*, onde se efectua todo o processamento de dados, e pela classe *ProgramOptions* que não é mais do que uma estrutura onde estão armazenadas as opções do programa. Para além disso, possui a classe *Genome* que inclui todos os atributos e resultados do processamento de um determinado genoma, inclusive uma instância de *Chromosome* que inclui por sua vez todos os atributos e resultados do processamento de um determinado cromossoma.

Por último, o módulo *data* abrange todas as funcionalidades relativas a dados, desde a estrutura de ficheiros escolhidos pelo utilizador (*FileList*), passando pelas classes *AGenDAFileFilter* e *GenomeFileFilter* que filtram o tipo de ficheiros a seleccionar, a classe *BigFile* que implementa a interface *Iterable* que permite atravessar todos os elementos de uma colecção, até à classe *DataGenerator* que cria ficheiros onde são armazenados os resultados obtidos.

3.2.2 Estruturas de dados

Uma vez que estamos a lidar com elevada quantidade de informação, é importante criar estruturas de dados que a armazene de uma forma organizada.

A escolha das estruturas de dados a usar depende de diversas questões, nomeadamente da forma como se pretende guardar e aceder aos dados, nas dimensões que ela irá atingir e na sua rapidez. Por este motivo, serão usadas apenas duas estruturas diferentes. Nos casos em que se conhece *a priori* o tamanho da estrutura que vamos utilizar e não são necessárias inserções nem remoções constantes de elementos, será usado um *array* estático por ser o tipo de estrutura mais rápido. No entanto, em muitas situações não se sabe o tamanho inicial da estrutura e este pode ir aumentando ao longo do tempo, sendo por isso necessário usar estruturas dinâmicas. Nestas situações será usado o *HashMap*, uma *Collection* do Java que implementa a interface *Map*. Esta estrutura utiliza chaves (*keys*) para aceder a posições de memória, que podem ser palavras (*String*), o que mostrou ser vantajoso no armazenamento das distâncias, uma vez que podemos associar cada *key* a uma determinada palavra. O *HashMap* é essencialmente uma implementação do *HashTable* sem sincronização interna, sacrificando essa característica para obter um maior desempenho. Como o processamento das distâncias será efectuado num único *thread*, esta é sem dúvida a melhor opção a utilizar devido à sua rapidez.

Relativamente à forma como os ficheiros e as pastas se encontram organizados, há que ter em conta que ao seleccionar os ficheiros que se pretende processar o utilizador está já a definir a maneira como estes serão armazenados e consequentemente processados, isto de maneira a facilitar o tipo de análise que se pretende fazer.

O objectivo da ferramenta é analisar as distâncias inter-simbólicas de dados genéticos, dados estes que estão normalmente divididos em vários ficheiros que contêm as sequências de nucleótidos de um cromossoma específico. Assim, partindo do pressuposto de que um ficheiro equivale a um cromossoma, podemos então considerar que uma pasta com vários ficheiros equivale a um genoma de uma espécie. Logo, quando se pretender fazer um estudo comparativo entre diferentes genomas é conveniente criar uma pasta diferente para cada genoma, onde estarão os ficheiros com os cromossomas respectivos. No entanto, este tipo de comparação não é a único possível de efectuar com a ferramenta, a ideia é que também seja possível fazer comparações entre genomas ou cromossomas. Por isso, caso se pretenda comparar cromossomas é conveniente criar uma pasta para cada cromossoma e desta forma será feita a comparação entre eles. De uma maneira generalizada, pode dizer-se que os estudos comparativos serão feitos baseados na divisão entre as diferentes pastas na área de trabalho.

Contudo, para facilitar o desenvolvimento da ferramenta ficou estipulado que cada pasta é considerada um genoma e cada ficheiro dentro de uma pasta é considerado um cromossoma desse mesmo genoma (Figura 3.3), fazendo uso das classes *Genome* e *Chromosome* que já foram apresentadas na secção anterior. É de salientar que esta consideração é relativa apenas ao nome dado às classes, continuando a ser possível fazer comparações entre cromossomas se estes forem colocados em diferentes pastas.

A classe *Genome* é então usada para armazenar a informação de um genoma. Esta informação inclui o nome do genoma, ou seja, o nome da pasta na árvore de directórios, um *array* de cromossomas e um conjunto de dados necessários para o processamento das distâncias e obtenção de resultados. Esses dados estão divididos em dados individuais e dados conjuntos. Os dados conjuntos referem-se à totalidade das frequências, ou seja, a todas as frequências das distâncias de todas as palavras. Enquanto que os dados individuais referem-se à frequência das distâncias de uma única palavra de cada vez. Para além dos acima descritos, os atributos da classe *Genome* são os seguintes:

- **finalConjointAbsoluteFreqByFile**: é um *array* em que cada posição está associada a um ficheiro (cromossoma) e contém a frequência absoluta de cada distância desse ficheiro;
- **finalConjointRelFreqByFile**: é um *array* em que cada posição está associada a um ficheiro (cromossoma) e contém a frequência relativa de cada distância desse ficheiro;
- **finalConjointAbsoluteFreq**: contém a frequência absoluta de cada distância da totalidade dos ficheiros;
- **finalConjointRelFreq**: contém a frequência relativa de cada distância da totalidade dos ficheiros;
- **finalRefConjointRelFreq**: contém a distribuição de referência de cada distância da totalidade dos ficheiros;
- **finalIndividualAbsoluteFreqByFile**: é um *array* em que cada posição está associada a um ficheiro (cromossoma) e contém a frequência absoluta de cada distância de cada palavra desse ficheiro;
- **finalIndividualAbsoluteFreq**: contém a frequência absoluta de cada distância de cada palavra da totalidade dos ficheiros;
- **finalIndividuallRelFreq**: contém a frequência relativa de cada distância de cada palavra da totalidade dos ficheiros;
- **seqOccur**: contém a totalidade da frequência de distâncias de cada palavra de todos os ficheiros;
- **totalOccur**: contém a totalidade da frequência de distâncias de todos os ficheiros.

A classe *Chromosome* é usada para armazenar a informação de um cromossoma. Esta informação engloba o nome do cromossoma, ou seja, o nome do ficheiro na árvore de directórios, o caminho onde se localiza o ficheiro no disco e também alguns dados necessários para o processamento das distâncias:

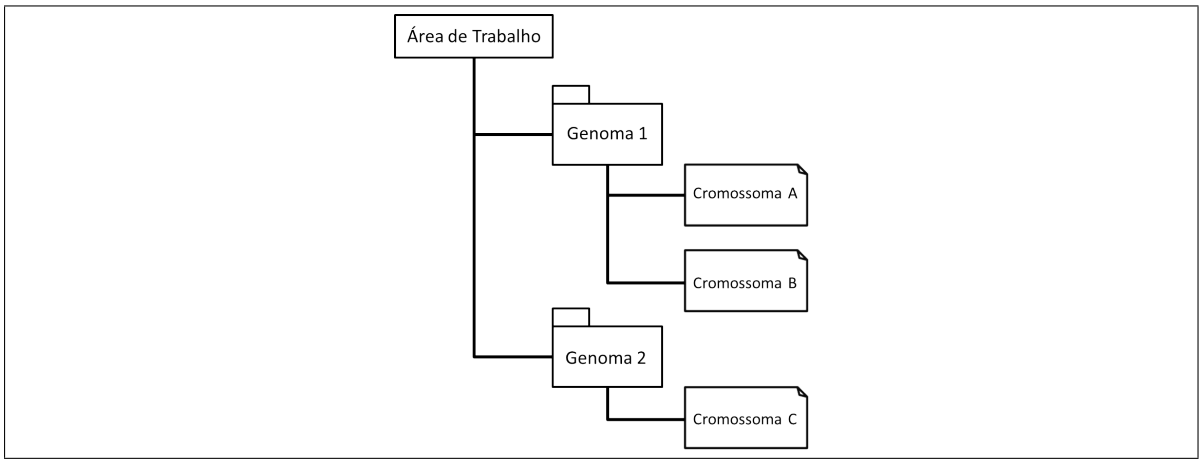


Figura 3.3: Estrutura de directórios da área de trabalho.

- **distanceFrequencyByFrame**: é um array em que cada posição está associada a uma *reading frame* e contém a frequência das distâncias de cada palavra dessa *reading frame*;
- **wordFrequencyByFrame**: é um array em que cada posição está associada a uma *reading frame* e contém a frequência total de cada palavra dessa *reading frame*;

3.3 Implementação

A implementação consiste na realização propriamente dita da ferramenta, utilizando como base a arquitectura definida na secção anterior e centrando-se nos objectivos definidos pelos requisitos funcionais e não funcionais.

Nesta secção será então apresentada a linguagem de programação escolhida para o desenvolvimento da ferramenta, assim como as bibliotecas complementares utilizadas. De seguida, serão descritos os dois modelos de análise usados, os métodos quantitativos e as representações gráficas dos dados. Serão também explicados alguns algoritmos e o seu processamento, e será descrito o fluxo de dados. Para terminar a descrição da implementação da ferramenta, será apresentada a interface gráfica onde se poderão ver todas as funcionalidades da mesma.

3.3.1 Linguagem e ferramentas de desenvolvimento

A ferramenta AGenDA foi desenvolvida na linguagem Java, em grande parte pelo facto de esta ter a característica de se escrever uma vez e executar em qualquer lugar (*"write once run anywhere"*), tal como a *Sun Microsystems*, empresa que a desenvolveu, a descreve. Como esta linguagem é compilada para um código binário, que por sua vez é executado numa máquina virtual, basta ter a *Java Virtual Machine* (JVM) instalada para conseguir executar um programa em Java, sendo portanto compatível com a grande maioria dos sistemas operativos. Para além disso, é uma linguagem bastante rápida que incorpora o *Swing*, uma API que possibilita a criação de interfaces gráficas muito completas, tendo-se mostrado por isso a opção mais adequada para o desenvolvimento da ferramenta.

Foram também usadas bibliotecas complementares, como a *JFreeChart* [12] para a criação de gráficos e histogramas, e a *MigLayout - Java Layout Manager* para a criação de uma interface gráfica com um *layout* personalizado.

JFreeChart é uma biblioteca que permite desenhar gráficos de diversos tipos e suporta várias funcionalidades como *tooltips* e ferramentas de zoom. É relativamente fácil de utilizar, usando sempre o mesmo critério para a construção de qualquer tipo de gráfico. O primeiro passo consiste em criar um conjunto de dados, utilizando a classe *DataSet*, que será então usado na criação do gráfico propriamente dito. O objecto que representa os gráficos tem o mesmo nome que a biblioteca, *JFreeChart* e é construído através da classe *ChartFactory*. A Figura 3.4 ilustra um exemplo de como criar um gráfico circular (*pie chart*) através de um conjunto de dados previamente criados (*pie data set*).

```
DefaultPieDataset pieDataset = new DefaultPieDataset();
pieDataset.setValue("Linux", 29);
pieDataset.setValue("Mac", 20);
pieDataset.setValue("Windows", 51);
JFreeChart chart = ChartFactory.createPieChart
                    ("Sample Pie Chart",
                     pieDataset,
                     true
                    );
```

Figura 3.4: Exemplo da criação de um gráfico com a biblioteca *JFreeChart*.

O *MigLayout* é um *Layout Manager* muito flexível que fornece a possibilidade de criar *layouts* personalizados, sem qualquer restrição. Esta biblioteca foi fundamental na construção da interface gráfica pretendida, uma vez que os *layouts* existentes na API *Swing* são bastante limitados.

Uma das formas de criar um *layout* personalizado, consiste em ir adicionando novos elementos a um *Container* como se este fosse uma grelha. Neste caso, ao adicionar um elemento usando o método *add* ele será posicionado numa nova coluna na mesma linha. Quando se pretender passar para a linha seguinte, será necessário adicionar a palavra *wrap*. Assim, o último elemento ocupará o resto da linha onde se encontra e o próximo será colocado numa nova linha. Na Figura 3.5 podemos ver um exemplo dessa utilização, onde é inserida a palavra *wrap* no momento em que se adiciona o terceiro elemento, o que faz com que o quarto fique posicionado na linha seguinte.

```
JPanel panel = new JPanel(new MigLayout());
panel.add(comp1);
panel.add(comp2);
panel.add(comp3, "wrap")
panel.add(comp4);
```

comp1	comp2	comp3
comp4		

Figura 3.5: Exemplo de como adicionar elementos por ordem usando a biblioteca *MigLayout*.

Partindo deste pressuposto, há depois uma enorme variedade de acções que se podem fazer com as células, desde dividir uma célula em várias (através do *split*), fundir várias células numa só (usando *span*), etc. A Figura 3.6 exemplifica dois usos diferentes do *span*, num é definido o tamanho do mesmo ("*span* 2") o que faz com que o segundo elemento ocupe duas células. No outro não é definido o tamanho, fazendo com que o quarto elemento ocupe a linha inteira.

```
panel.add(comp1);
panel.add(comp2, "span 2");
panel.add(comp3, "wrap");
panel.add(comp4, "span");
```

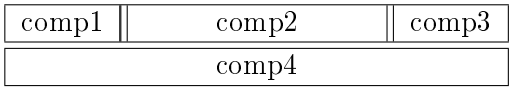


Figura 3.6: Exemplo de como dividir ou fundir células usando o *MigLayout*.

Também existe a possibilidade de especificar logo na declaração quantas colunas pretendemos que a grelha tenha e, desta forma, não será necessário utilizar a palavra *wrap* para efectuar a mudança de linha (por exemplo, *new MigLayout("wrap 3")*).

Outra maneira de definir o *layout* pretendido é adicionando elementos através das coordenadas absolutas. Desta forma, podemos definir posições específicas sem ter de seguir a ordem da grelha. Na Figura 3.7 é exemplificada esta utilização. Numa das linhas são também acrescentados dois valores a seguir às coordenadas, que definem a largura e a altura da célula (*cell <column> <row> <width> <height>*). Em qualquer uma destas abordagens descritas, podem ainda ser usados outros parâmetros que definem uma série de características, por exemplo tamanhos, alinhamentos e espaçamentos (*width <min>:<preferred>:<max>*, *align <left>*, *gaptop <2>*).

```
panel.add(comp1, "cell 0 0");
panel.add(comp2, "cell 1 0 2 1");
panel.add(comp3, "cell 3 0");
panel.add(comp4, "cell 0 1 4 1");
```

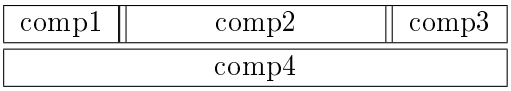


Figura 3.7: Exemplo de como adicionar elementos através das coordenadas absolutas usando o *MigLayout*.

Assim, através de todas as funcionalidades desta biblioteca é possível criar qualquer *layout* que se pretenda, por mais complexo que ele seja.

3.3.2 Modelos de análise

Uma vez que a estrutura do ADN é muito complexa e contém diferentes regiões, o seu tipo de análise pode variar dependendo daquilo que se pretende explorar. Quando se analisa as regiões codificantes, por exemplo, poderá ser conveniente considerar uma *reading frame* que extrai palavras de uma maneira muito semelhante à que os codões são extraídos de uma sequência. No entanto, nalguns casos esta poderá não ser a melhor solução, pois algumas distâncias não são consideradas e consequentemente há perda de informação.

Por este motivo, foram desenvolvidas duas abordagens diferentes no cálculo das distâncias entre oligonucleótidos: uma que considera L *reading frames*, onde L é o comprimento da palavra, e outra que procura todas as ocorrências de palavras de comprimento L sem considerar a existência de *reading frames*.

A primeira abordagem divide a sequência em grupos de palavras de tamanho L, começando em L diferentes pontos da sequência para construir as diferentes *reading frames*. Esta abordagem foi baseada no estudo dos dinucleótidos [4] e estendida a palavras de qualquer

tamanho. Existem então L possíveis *reading frames*: a *reading frame* 1 ($R1$) que começa no primeiro nucleótido da sequência, a *reading frame* 2 ($R2$) que começa no segundo nucleótido e assim sucessivamente até criar L *reading frames* diferentes, em que L é o tamanho da palavra. É de salientar que, para palavras de tamanho um, obtêm-se os mesmos resultados quer se utilize a abordagem com ou sem *reading frames*. Assim, iremos ilustrar a abordagem com *reading frames* utilizando palavras de tamanho dois e considerando a seguinte sequência:

$$AATTTGTATTCTTAAACAAATTC$$

Ao analisarmos as distâncias entre dinucleótidos ($L = 2$) existem duas *reading frames* diferentes, agrupadas da seguinte forma:

$$R1 : \{AA, TT, TG, TA, TT, CT, TA, AA, CA, AA, TT\}$$
$$R2 : \{AT, TT, GT, AT, TC, TT, AA, AC, AA, AT, TC\}$$

A sequência de distâncias de cada palavra é um vector que contém as distâncias entre consecutivas ocorrências dessa palavra. Considera-se ainda que a sequência simbólica é cíclica, isto é, o último nucleótido é concatenado ao primeiro para formar outra palavra caso ainda não tenha sido usado. Por exemplo, na $R1$ o último C não é concatenado com o primeiro A pois este já foi utilizado para a primeira palavra AA. O alfabeto de nucleótidos é $\{A, C, G, T\}$, por isso neste caso existem 16 (4^L) palavras e consequentemente 16 diferentes sequências de distâncias. Para a sequência fornecida no exemplo anterior, as duas *reading frames* da sequência de distância da palavra AA e TT são:

$$d_{R1}^{AA} = (7, 2, 2), \quad d_{R1}^{TT} = (3, 6, 2)$$
$$d_{R2}^{AA} = (2, 9), \quad d_{R2}^{TT} = (4, 7)$$

É também possível construir uma sequência conjunta de distâncias entre oligonucleótidos, constituída por todas as distâncias das diferentes palavras no mesmo vector. O comprimento desta sequência é igual à soma dos comprimentos das 16 sequências individuais de distâncias. Desta forma, as sequências conjuntas de distâncias das duas *reading frames* são:

$$d_{R1} = (7, 3, 11, 3, 6, 11, 8, 2, 11, 2, 2)$$
$$d_{R2} = (3, 4, 11, 6, 6, 7, 2, 11, 9, 2, 5)$$

Na segunda abordagem, não se utilizam as *reading frames*. Assim, em vez de se considerarem múltiplas *reading frames* que agrupam nucleótidos, começa-se por extrair a primeira palavra da sequência e de seguida procura-se a sua próxima ocorrência. Como exemplo ilustrativo, usando a sequência anterior e palavras também de tamanho dois, a sequência das distâncias sem *reading frame* seria:

$$d^{AA} = (12, 3, 2, 5, 4), \quad d^{TT} = (5, 4, 2, 8, 4)$$

Neste caso, a sequência conjunta de distâncias seria então:

$$d = (12, 5, 5, 4, 22, 22, 11, 2, 11, 22, 8, 16, 3, 2, 22, 5, 5, 4, 4, 4, 10)$$

É de salientar que as distâncias processadas nestas duas abordagens encontram-se em unidades diferentes. Na primeira, uma unidade de distância corresponde a uma palavra de

tamanho N, ou seja, a uma *reading frame*. Na segunda, uma unidade de distância corresponde a um único nucleótido.

Para além destas duas abordagens que definem um modo diferente de análise das distâncias, foi ainda criado um outro modelo que pode ser usado em regiões codificantes. A diferença deste modelo em relação ao anterior é que está preparado para extrair as diferentes regiões do ficheiro, começando cada nova região com uma nova contagem de distâncias.

3.3.3 Métodos quantitativos e visualização gráfica de dados

De maneira a fazer uma melhor interpretação dos resultados obtidos através do processamento das distâncias inter-simbólicas, foram utilizadas algumas visualizações gráficas de dados baseadas no cálculo de métodos quantitativos.

É então utilizada a distribuição conjunta e individual das distâncias entre oligonucleótidos. Assim, a distribuição conjunta proporciona uma visão geral de todas as distâncias, enquanto que a distribuição individual refere-se exclusivamente às distâncias de uma determinada palavra (oligonucleótido). Existindo portanto uma distribuição conjunta e 4^L distribuições individuais diferentes, onde L é o tamanho da palavra. De maneira a visualizar estas distribuições são criados histogramas a partir das frequências relativas das distâncias.

Por forma a estudar algumas propriedades estatísticas, é também criada uma distribuição de referência, tal como foi adoptada em dois estudos precedentes [1, 4]. Considera-se então que as sequências de nucleótidos foram geradas por um processo aleatório independente e igualmente distribuído (i.i.d.), sendo a sua função de distribuição definida por:

$$F^x(k) = P(d^x \leqslant k) = 1 - (1 - p^x)^k$$

com um valor esperado de $1/p^x$ e uma variância de $(1 - p^x)/(p^x)^2$.

Com o objectivo de extrair o fundo aleatório e adquirir informação sobre a evolução selectiva, é calculado o erro relativo definido por:

$$r(k) = \frac{f_0(k) - f(k)}{f_0(k)}$$

onde $f_0(k)$ é a frequência relativa da distância k observada e $f(k)$ é a função massa de probabilidade associada a um processo aleatório i.i.d.. Esta análise salienta as características selectivas da evolução do ADN de cada espécie permitindo descobrir semelhanças entre elas.

Também é utilizada a divergência de *Kullback-Leibler* com o indicador de semelhança entre duas funções de probabilidade. A divergência de *Kullback-Leibler*, para duas funções de probabilidade g e h, é definida da seguinte forma:

$$D_{KL}(g\|h) = \sum_i g(i) \log \frac{g(i)}{h(i)}$$

Esta medida de divergência não é simétrica, possui sempre valores positivos ou nulos e apenas é nula quando $g = h$.

Assim, utilizando o cálculo da divergência de *Kullback-Leibler* são criadas tabelas que proporcionam informação sobre as diferenças entre a distribuição das distâncias inter-simbólicas.

3.3.4 Algoritmos e processamento de dados

O processamento de dados apenas poderá ocorrer assim que se verifiquem todos os pré-requisitos necessários à sua realização. Isso implica que já existam ficheiros prontos a ser analisados, que os modelos de análise tenham sido especificados pelo utilizador e que este solicite o início dos mesmos.

Uma vez que se pretende criar uma interface gráfica que permita a interacção entre o utilizador e o programa, é fundamental garantir que esta não bloqueie e que esteja sempre acessível para ser usada a qualquer instante. Como já foi referido, optou-se pelo uso da API *Swing* do Java para a criação da interface gráfica, que será descrita mais à frente na secção 3.3.6. Os componentes do *Swing* não suportam o acesso de múltiplos *threads* e ocorrem normalmente no *Event Dispatch Thread* (EDT) [39]. Isto significa que nenhuma acção ocorre em paralelo, o que pode trazer problemas caso exista uma acção mais demorada que ocupe o *thread* por muito tempo e consequentemente bloqueie a interface. Por este motivo, é conveniente considerar diferentes *threads* ao criar a interface gráfica. Assim, foi adoptada a abordagem mais comum que consiste em utilizar um *thread* que inicializa os elementos constituintes da interface (*initial thread*), outro que executa os eventos internos dos elementos (EDT) e outro que executa o processamento de tarefas mais longas (*worker thread*). O *initial thread* limita-se a agendar a criação dos elementos iniciais da interface através do método estático *invokeLater* da classe *SwingUtilities*. A partir do momento em que é inicializado, a aplicação passa a correr baseada nos eventos gerados pelo utilizador, tais como carregar num botão ou abrir uma janela, que são executados no EDT. No *worker thread* será então executada a tarefa mais longa, que consiste no processamento das distâncias. Para isso, é implementada uma subclasse da classe abstracta *SwingWorker*, que permite executar tarefas em *background*. Esta subclasse implementa o método *construct()* que contém a tarefa longa a efectuar e ainda o método *finished()* que será executado na EDT assim que o método *construct()* terminar.

Desta forma, é possível criar uma interface que responde às acções feitas pelo utilizador mesmo quando está a decorrer o processamento de dados, permitindo até cancelar o *worker thread* e interromper o seu processamento. É de salientar, que o processamento dos dados será sempre efectuado num único *thread*, não sendo necessário utilizar estruturas de dados com sincronização interna, como já foi referido.

Uma vez resolvida a questão do início do processamento, podemos então prosseguir para a descrição detalhada do mesmo. A estrutura de um ficheiro FASTA consiste, como já foi explicado, numa sequência de letras em que cada letra equivale a um nucleótido, podendo também existir símbolos indefinidos. De maneira a simplificar o processo, todos os símbolos indefinidos foram previamente eliminados e só depois foi feito o processamento dos dados.

A abordagem utilizada para o estudo das distâncias inter-simbólicas, consistiu em dividir o processamento em várias etapas, como se pode ver na Figura 3.8. Numa primeira instância, cada ficheiro de nucleótidos é analisado e é feito o cálculo da frequência de cada distância para cada oligonucleótido e é calculado também o número total de ocorrências. Estes valores são armazenados em estruturas de dados locais do tipo *HashMap*, descritas na secção 3.2.2, que tem como chave a palavra e como valor um outro *HashMap*. Nesse *HashMap*, por sua vez, a chave corresponde à distância e o valor ao número de ocorrências dessa distância. A tabela 3.1 ilustra um exemplo dessa estrutura considerando uma distância máxima de 6 e palavras de tamanho 1 (A, G, C e T).

Estes dados são depois associados às estruturas internas de cada cromossoma, armazenando assim informação sobre a frequência das distâncias entre palavras e o total de ocorrências. Es-

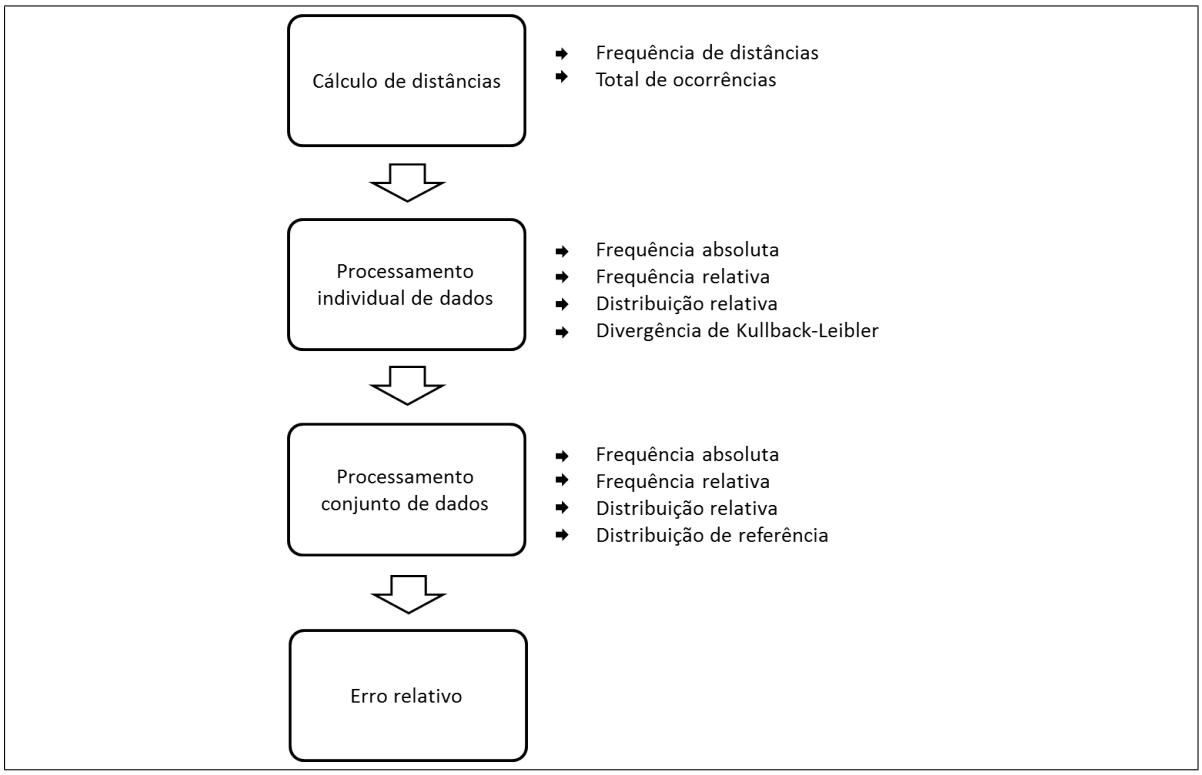


Figura 3.8: Esquema das etapas de processamento do programa.

tas duas estruturas de dados são *arrays* estáticos uma vez que, quando se utiliza a abordagem de *reading frame*, existirá uma frequência de distâncias para cada *frame*. Estes *arrays* terão tamanho L para palavras de comprimento L, uma vez que o número de *reading frames* é igual ao comprimento da palavra. Caso se utilize a bordagem sem *reading frame* o tamanho será 1.

A segunda etapa consiste em fazer o processamento de dados individual, isto é, fazer os cálculos estatísticos para cada palavra individualmente. Estes cálculos incluem a frequência absoluta, frequência relativa para posterior cálculo do perfil de erros relativos, e também a divergência de *Kullback-Leibler*.

A frequência individual absoluta das distâncias corresponde à soma total das frequências

Tabela 3.1: Frequência das distâncias entre nucleótidos da espécie *Vitis vinifera* para palavras de tamanho um.

A		C		G		T	
Distância	Ocorrências	Distância	Ocorrências	Distância	Ocorrências	Distância	Ocorrências
1	5.801×10^8	1	1.811×10^8	1	1.812×10^8	1	5.787×10^8
2	3.041×10^8	2	1.203×10^8	2	1.200×10^8	2	3.035×10^8
3	2.016×10^8	3	0.971×10^8	3	0.970×10^8	3	2.012×10^8
4	1.331×10^8	4	0.698×10^8	4	0.698×10^8	4	1.333×10^8
5	0.934×10^8	5	0.575×10^8	5	0.576×10^8	5	0.933×10^8
6	0.652×10^8	6	0.490×10^8	6	0.490×10^8	6	0.653×10^8

individuais de todas as *reading frames*. Caso não se utilize o processamento com *reading frames* não é necessário fazer este cálculo, pois existe apenas uma única *reading frame* sendo esta já considerada a frequência individual absoluta das distâncias.

De seguida, é calculada a frequência individual relativa utilizando as frequências individuais absolutas e dividindo cada frequência de cada palavra pelo número total de ocorrências dessa palavra, da forma descrita no Algoritmo 3.3.1. A partir da frequência relativa calculada, são então processadas as distribuições individuais relativas na forma de histogramas.

Algoritmo 3.3.1: INDIVIDUALRELFREQ($valueX, valueY$)

---

while $iterSeq.hasnext()$

do { **while** $iterDist.hasnext()$

do { $aux \leftarrow individualAbsoluteFreq.get(seq).get(dist)/seqOccur.get(seq)$;
$individualRelFreq.get(seq).put(dist, aux)$;

O cálculo da divergência de *Kullback-Leibler*, foi feito através da expressão fornecida na secção 3.3.2. Este cálculo utiliza um par de valores em cada ciclo (Algoritmo 3.3.2), sendo que esses valores são obtidos da estrutura de dados que contém a frequência relativa individual das distâncias. Assim, cada valor da função *KLvalue* consiste na comparação entre um par de palavras da frequência relativa, obtendo-se desta forma uma tabela $4^L \times 4^L$, em que L é o tamanho da palavra. Esta divergência é calculada considerando um número de distâncias igual à distância máxima definida *a priori*. Por exemplo, para a distância máxima 100, a soma total das frequências relativas das 100 distâncias deve ser 1. No entanto, uma vez que a frequência relativa é calculada considerando todas as distâncias existentes não é garantido que esta soma seja um. Por esse motivo, os valores das 100 primeiras distâncias precisam de ser normalizados. Assim, todos os valores são divididos pelo valor da normalização ($normX$ e $normY$).

A etapa seguinte consiste em processar os dados conjuntos, que contêm as frequências das distâncias de todas as palavras. Para este cálculo é primeiro necessário somar todas as distâncias de todas as palavras, de maneira a obter a frequência conjunta absoluta. Esta frequência é uma estrutura de dados em que cada índice corresponde a uma distância e contém o número de vezes que ocorreu essa distância em qualquer palavra. Posteriormente, é feito o cálculo da frequência conjunta relativa de uma forma semelhante à que foi calculada a frequência individual relativa e da mesma forma são criados os histogramas com as distribuições conjuntas relativas.

Por último, é calculado o erro relativo. Embora seja usado o conjunto de dados para obter o erro relativo, o seu processamento é feito em separado, daí ser considerada uma etapa diferente. Assim, depois de todo o processamento conjunto ser efectuado, é criado o erro relativo conjunto de cada genoma (directório) de maneira a serem inseridos no mesmo gráfico.

A expressão usada para este cálculo encontra-se descrita na secção 3.3.2.

Algoritmo 3.3.2: KLVALUE($valueX, valueY$)

$res \leftarrow 0$
for $i \leftarrow 0$ **to** lim
do
- **if** $valueX[i] = null$
  - **then** $aux1 \leftarrow 0$
  - **else**
    - **if** $valueY[i] = null$
      - **then** $aux1 \leftarrow 0$
      - **else**
        - $aux1 \leftarrow (valueX[i]/normX)/(valueY[i]/normY)$
        - **if** $aux1$ **not** 0
          - **then** $res \leftarrow (valueX[i]/normX) \cdot \log(aux1)$

return (res)

3.3.5 Fluxo de dados

Uma vez descrito todo o processamento de dados efectuado pelo programa, resta ainda descrever de que forma são organizados e armazenados os resultados obtidos.

Como podemos ver através do fluxo de dados descrito na Figura 3.9, este programa tem como ponto de partida um ou mais ficheiros de dados FASTA que contêm as sequências de nucleótidos a analisar. Esses ficheiros são então processados e são calculadas as frequências de distâncias. Depois de terminado todo o processamento de dados, obtemos então quatro grupos de resultados: histogramas, gráficos, tabelas e dados. Os três primeiros estão disponíveis para visualização no próprio programa, enquanto que os dados contêm informação gerada pelo programa para calcular essas mesmas visualizações e apenas estão disponíveis para armazenar no disco do computador.

Os histogramas são distribuições relativas dos dados, distribuições essas que podem ser conjuntas ou individuais. Desta forma, serão então criados $4^L + 1$ histogramas, um dos quais com a distribuição conjunta relativa e os outros 4^L com as distribuições individuais relativas, em que L é o tamanho da palavra. Os erros relativos de todos os genomas processados são gerados num único gráfico e é também criada uma tabela com a divergência de *Kullback-Leibler* entre cada oligonucleótido para cada genoma. Os histogramas e os gráficos podem ser guardado no disco como imagem (*.png*), enquanto que as tabelas são guardadas em ficheiros de texto.

Os dados, que podem ser armazenados em ficheiros de texto (*.txt*), incluem a frequência individual absoluta de distâncias do genoma completo ou por cromossoma, e a frequência conjunta absoluta de distâncias do genoma completo ou por cromossoma. Todos os ficheiros de resultados possuem um cabeçalho com a descrição dos dados processados e das configurações de processamento usadas, como se pode ver através do exemplo da Figura 3.10. Estes dados podem depois ser utilizados noutros programas estatísticos por forma a obter outro valores ou visualizações gráficas não contempladas nesta ferramenta e assim permitir fazer outros estudos.

Para além disso, é ainda possível criar um ficheiro binário (*.AGenDA*) que pode ser carregado mais tarde pelo programa e gerar os resultados gráficos sem que seja necessário repetir todo o processamento de dados.

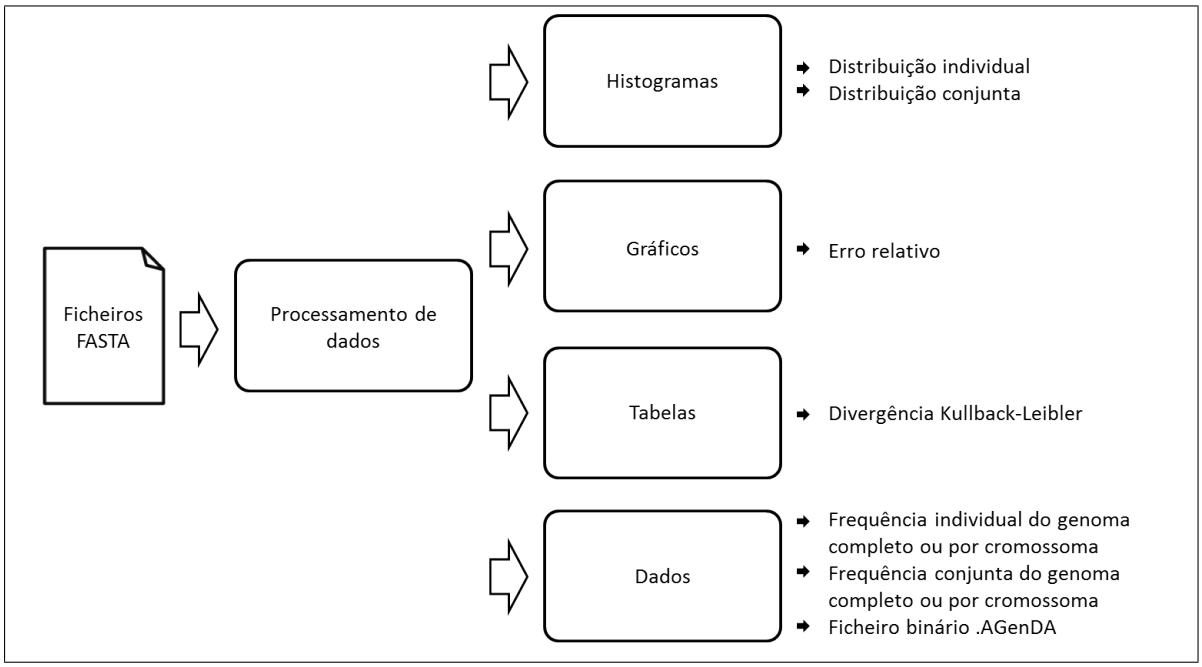


Figura 3.9: Fluxo de dados do programa.

```
################################################################################
#                                                                              #
#     Folder/Genome name:  e_colis            File/Chromosome name:  c7.fa     #
#                                                                              #
################################################################################
#                                                                              #
#     Options:  Reading frame?  No            Coding Regions?  No              #
#               Individual Analysis?  Yes     Conjoint Analysis?  Yes          #
#               Word size:  1                 Distance limit:  100             #
#                                                                              #
################################################################################
#                                                                              #
#     Type of study:  Conjoint Absolute Distance Frequency by Chromosome       #
#                                                                              #
################################################################################

   Seq.\ Dist.       1          2          3          4          5          6
        A        337870.0   178955.0   157179.0   104205.0   77486.0   70304.0
        C        271673.0   225179.0   208530.0   121855.0   87812.0   70587.0
        G        270137.0   223214.0   208326.0   121810.0   87758.0   71293.0
        T        339482.0   177135.0   157183.0   103515.0   77377.0   69395.0
```

Figura 3.10: Exemplo de ficheiro de resultados da frequência conjunta por cromossoma.

3.3.6 Interface gráfica

A criação de uma interface gráfica intuitiva foi considerada um aspecto fundamental desde o início do desenvolvimento desta ferramenta. Para isso, foram usados paradigmas familiares ao utilizador baseados nas ferramentas de análise e processamento de ADN existentes.

A estrutura da página principal do programa é bastante simples, consistindo numa janela que se encontra dividida em três partes principais: a área de trabalho, a área de resultados e a barra de menus (Figura 3.11).

A área de trabalho é constituída por dois separadores, o *Run FASTA File* e o *Load AGenDA*

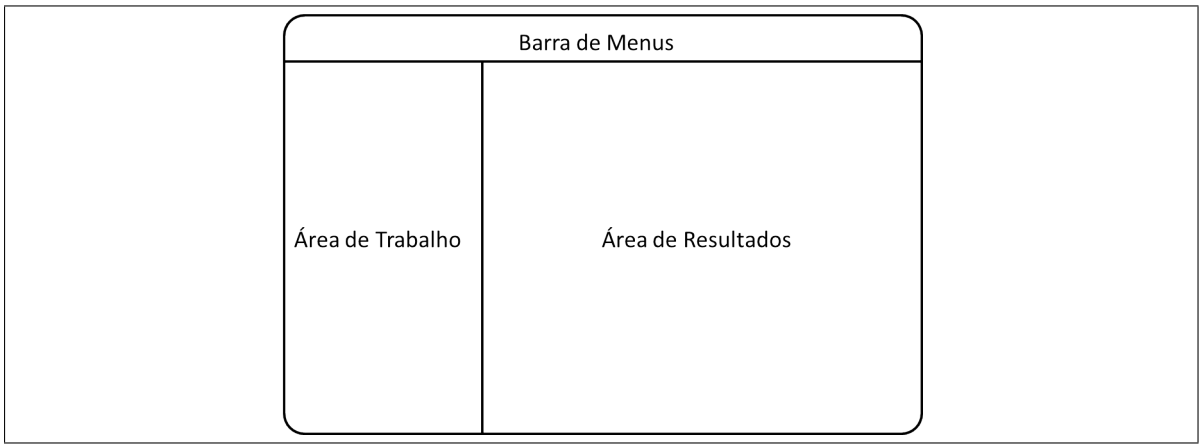


Figura 3.11: Estrutura da interface gráfica da ferramenta.

File. O primeiro separador permite carregar um ficheiro ou um directório de ficheiros FASTA para a árvore de directórios, sendo possível eliminar e adicionar mais ficheiros à área de trabalho (Figura 3.12a). Uma vez carregados os ficheiros, é possível escolher as diferentes opções de processamento (Figura 3.12b). Estas opções consistem em quatro caixas de selecção para escolher o tipo de análise (*Reading frame*, *Coding regions*, *Conjoint analysis* e *Individual analysis*) e dois *spinners* para ajustar os valores do tamanho da palavra e da distância máxima a considerar (*Word size* e *Max. dist.*)

De seguida, o utilizador tem de optar pelo tipo de processamento a fazer, se global ou local (Figura 3.12c). Para isso, existem dois painéis separados, um para o processamento global e outro para o local. O primeiro tem dois botões, o *Run selected* e o *Run all*, para processar apenas o ficheiro que estiver seleccionado ou todos os ficheiros na área de trabalho.

O segundo painel tem um único botão, o *Choose region*, que permite escolher uma região particular do ficheiro que estiver seleccionado. Ao pressionar este botão, surge uma nova janela constituída por dois separadores: o *Visual Selection* e o *Advanced Selection*. O *Visual Selection* mostra parte do ficheiro seleccionado numa caixa de texto (Figura 3.13a). Para além disso, contém um *spinner* onde se altera o número de linhas por página a mostrar, um botão que actualiza a caixa de texto, dois botões com setas para avançar e retroceder páginas e o botão *Run local* para iniciar o processamento das distâncias. No segundo separador da janela de análise local, existem vários *spinners* com parâmetros a definir pelo utilizador, uma imagem exemplificativa e o botão que inicia o processamento (Figura 3.13b).

O separador *Load AGenDA File* na área de trabalho, é constituído por um botão que serve para carregar um ficheiro, uma caixa de texto que mostra o nome do ficheiro carregado, outra um pouco maior que contém toda a informação do ficheiro carregado e um botão para processar o ficheiro.

Quando se carrega num dos botoes para processar os ficheiros, em qualquer uma das situações descritas, são mostrados os resultados gráficos na área de resultados. Estes resultados, por sua vez, estão divididos em três separadores: *Relative Frequency Histogram*, *Kullback-Leibler Divergence Tables* e *Relative Error*.

O primeiro separador (*Relative Frequency Histogram*), contém um *spinner* que serve para escolher o genoma a visualizar e dois botões de selecção que servem para mudar a vista para histograma conjunto ou histogramas individuais, como se pode ver na Figura 3.14. Quando se escolher a visualização dos histogramas individuais é ainda possível carregar nos botões

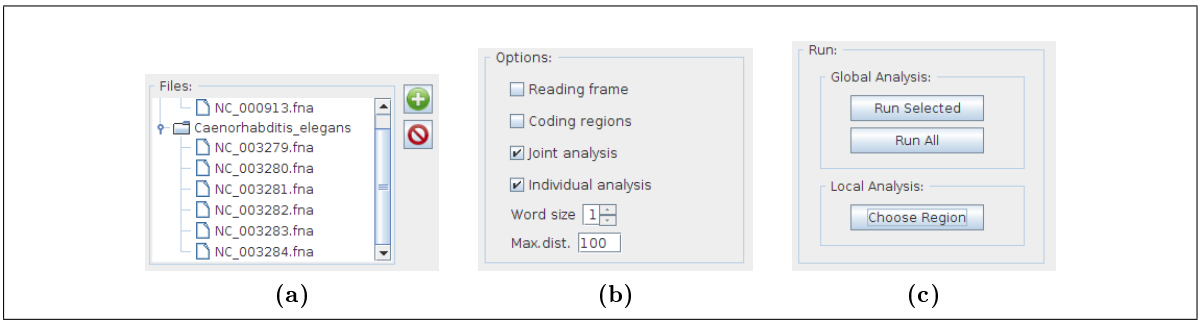


Figura 3.12: Área de trabalho.

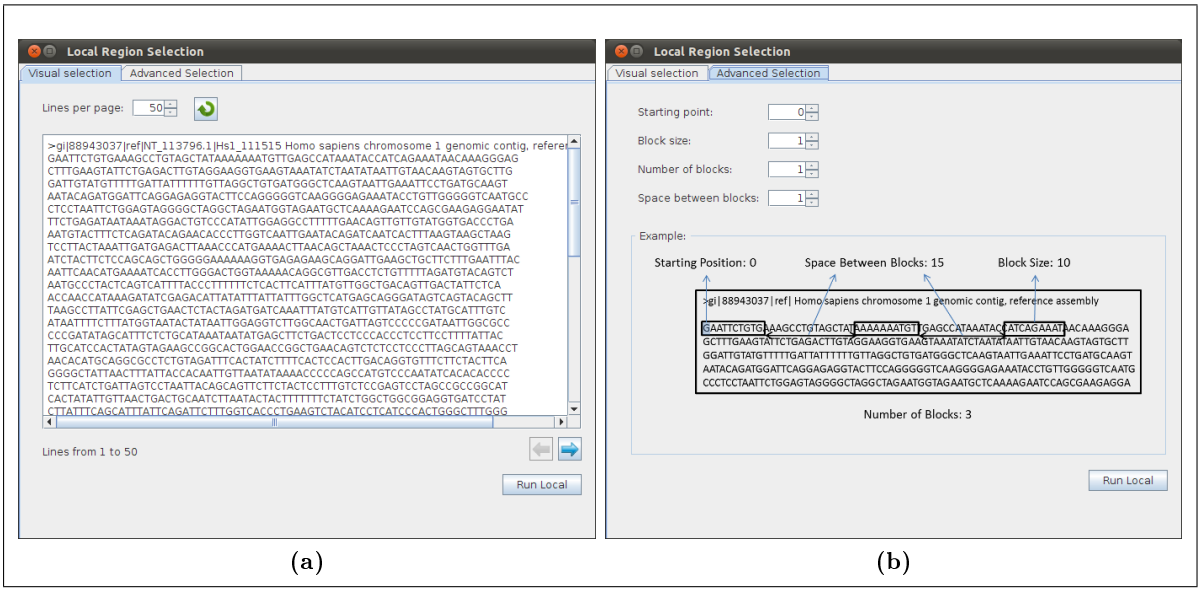


Figura 3.13: Separadores da janela de processamento local.

previous ou *next* para mudar para o histograma anterior ou seguinte, respectivamente.

O separador *Kullback-Leibler Divergence Tables* é constituído por uma tabela e um *spinner* que selecciona o genoma da tabela a apresentar (Figura 3.15).

O separador *Relative Error*, contém apenas o gráfico do perfil de erros relativos de todos os genomas processados (Figura 3.16).

Por fim, a barra de menus é constituída por três menus. O menu *File* contém as opções de abrir ficheiros FASTA carregar um ficheiro binário (.AGenDA), ou sair do programa. O menu *Tool* possui a opção de guardar os ficheiros de resultados. O menu *Help* permite consultar a documentação, o manual de utilizador ou a página informativa da ferramenta.

Quando se carrega no submenu *Save results*, surge uma nova janela que contém quatro separadores a dividir os tipos de dados que é possível guardar: *Histograms*, *Charts*, *Tables* e *Data* (Figura 3.17). O utilizador pode então seleccionar os resultados que pretende guardar e definir o directório onde serão guardados os ficheiros.

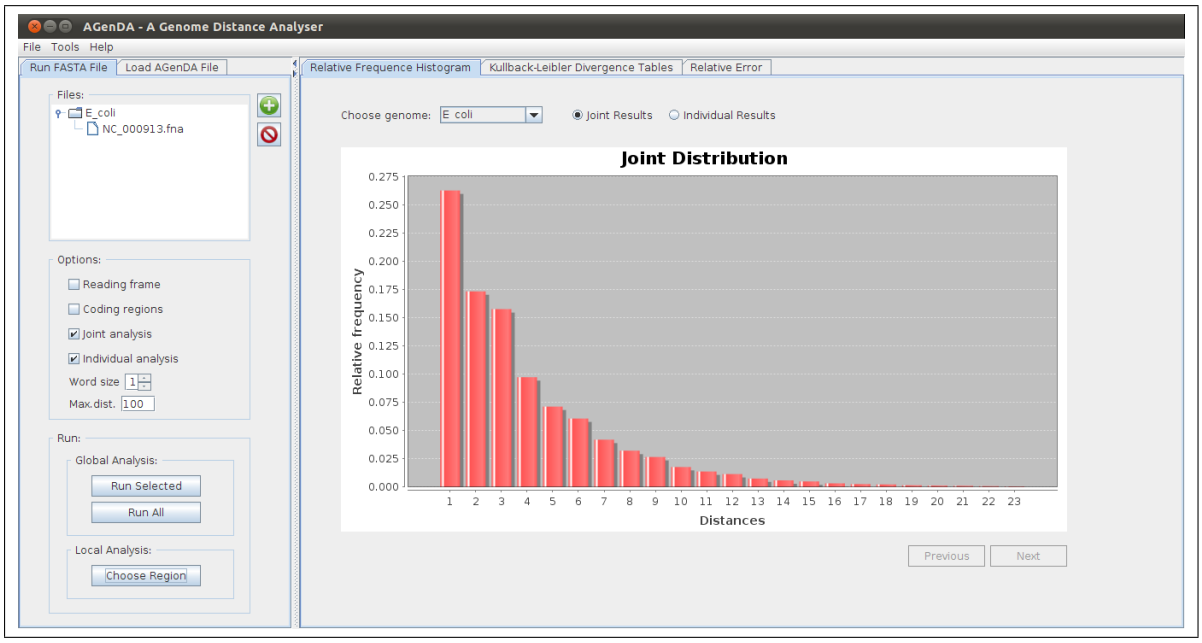
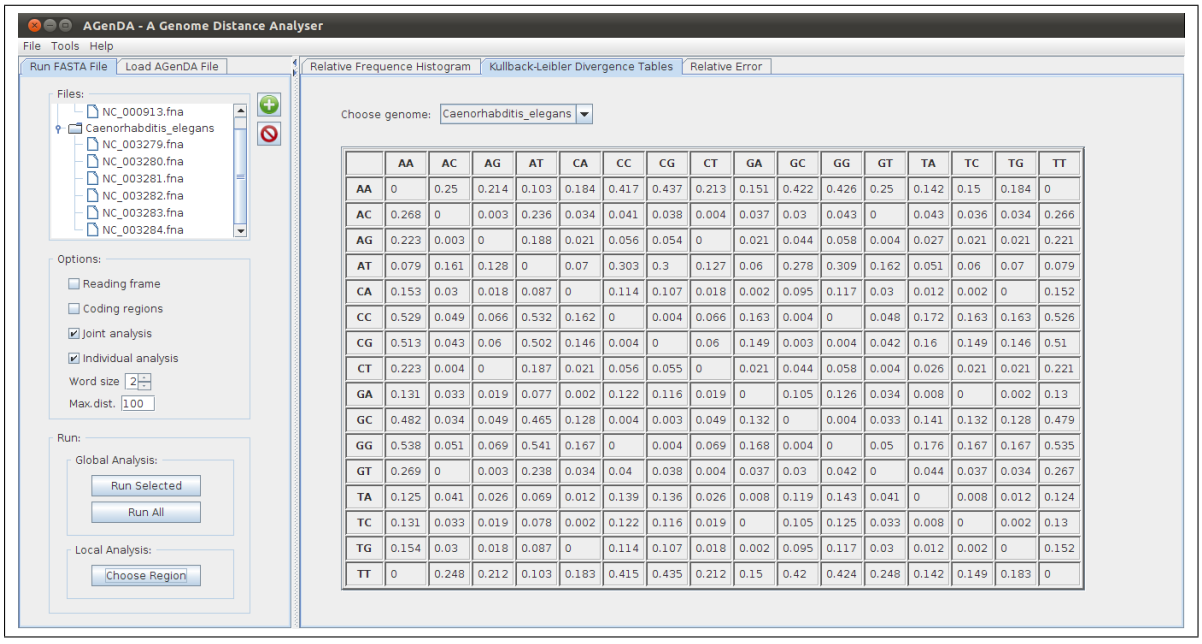


Figura 3.14: Separador dos histogramas da distribuição relativa da área de resultados.



	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
AA	0	0.25	0.214	0.103	0.184	0.417	0.437	0.213	0.151	0.422	0.426	0.25	0.142	0.15	0.184	0
AC	0.268	0	0.003	0.236	0.034	0.041	0.038	0.004	0.037	0.03	0.043	0	0.043	0.036	0.034	0.266
AG	0.223	0.003	0	0.188	0.021	0.056	0.054	0	0.021	0.044	0.058	0.004	0.027	0.021	0.021	0.221
AT	0.079	0.161	0.128	0	0.07	0.303	0.3	0.127	0.06	0.278	0.309	0.162	0.051	0.06	0.07	0.079
CA	0.153	0.03	0.018	0.087	0	0.114	0.107	0.018	0.002	0.095	0.117	0.03	0.012	0.002	0	0.152
CC	0.529	0.049	0.066	0.532	0.162	0	0.004	0.066	0.163	0.004	0	0.048	0.172	0.163	0.163	0.526
CG	0.513	0.043	0.06	0.502	0.146	0.004	0	0.06	0.149	0.003	0.004	0.042	0.16	0.149	0.146	0.51
CT	0.223	0.004	0	0.187	0.021	0.056	0.055	0	0.021	0.044	0.058	0.004	0.026	0.021	0.021	0.221
GA	0.131	0.033	0.019	0.077	0.002	0.122	0.116	0.019	0	0.105	0.126	0.034	0.008	0	0.002	0.13
GC	0.482	0.034	0.049	0.465	0.128	0.004	0.003	0.049	0.132	0	0.004	0.033	0.141	0.132	0.128	0.479
GG	0.538	0.051	0.069	0.541	0.167	0	0.004	0.069	0.168	0.004	0	0.05	0.176	0.167	0.167	0.535
GT	0.269	0	0.003	0.238	0.034	0.04	0.038	0.004	0.037	0.03	0.042	0	0.044	0.037	0.034	0.267
TA	0.125	0.041	0.026	0.069	0.012	0.139	0.136	0.026	0.008	0.119	0.143	0.041	0	0.008	0.012	0.124
TC	0.131	0.033	0.019	0.078	0.002	0.122	0.116	0.019	0	0.105	0.125	0.033	0.008	0	0.002	0.13
TG	0.154	0.03	0.018	0.087	0	0.114	0.107	0.018	0.002	0.095	0.117	0.03	0.012	0.002	0	0.152
TT	0	0.248	0.212	0.103	0.183	0.415	0.435	0.212	0.15	0.42	0.424	0.248	0.142	0.149	0.183	0

Figura 3.15: Separador das tabelas das divergências de *Kullback-Leibler* da área de resultados.

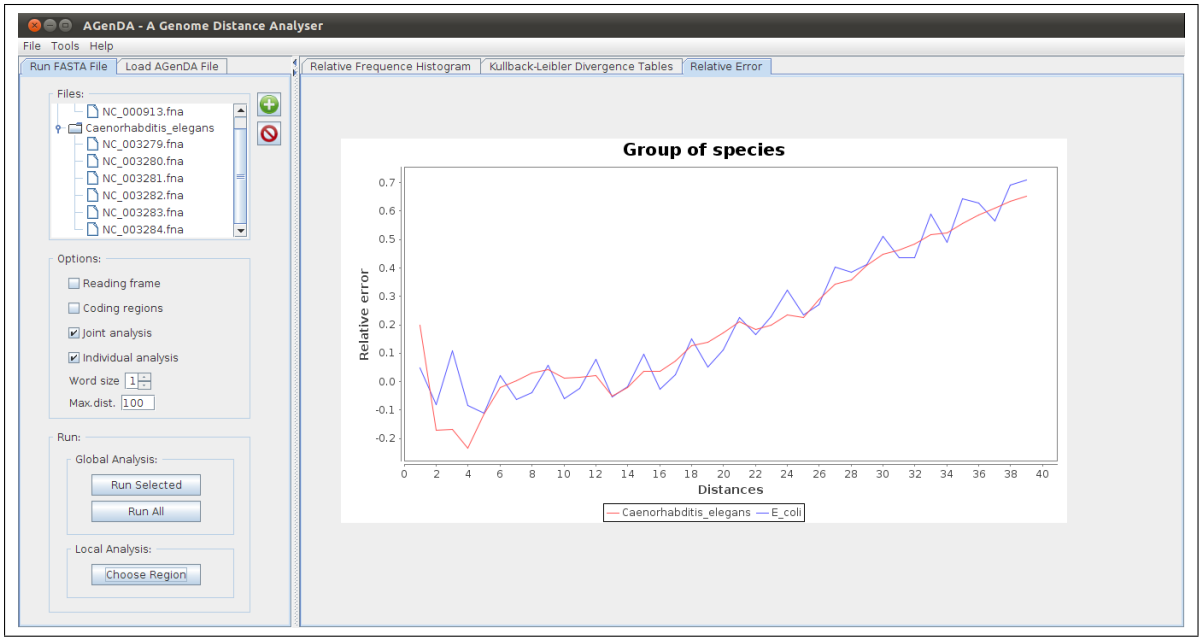


Figura 3.16: Separador dos gráficos do erro relativo da área de resultados.

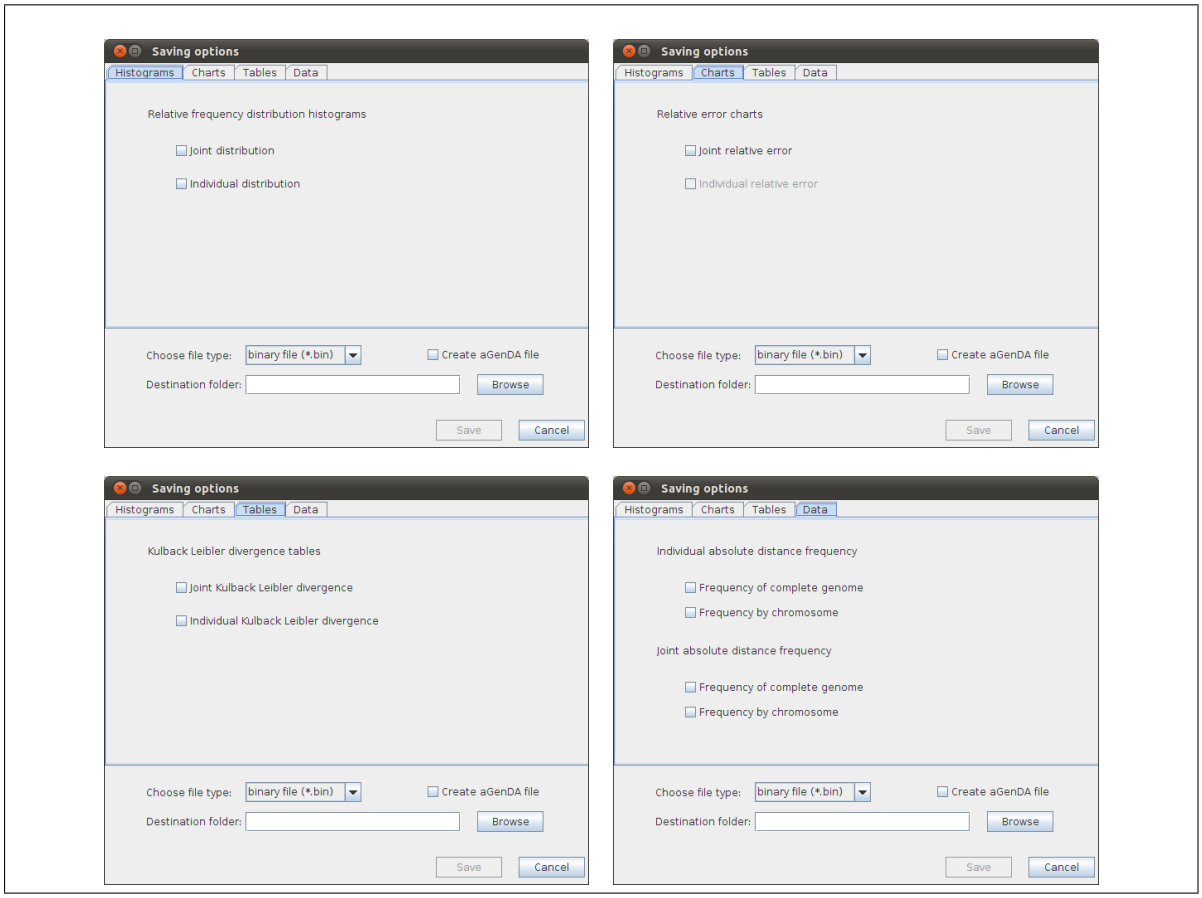


Figura 3.17: Janela das opções de resultados a guardar.

3.4 Documentação

A última etapa do desenvolvimento da ferramenta consistiu em criar um manual de utilização da mesma. Este manual tem como principal objectivo descrever detalhadamente as funcionalidades da ferramenta desenvolvida e explicar a sua utilização de uma maneira simples. Para isso, o documento foi dividido em cinco partes:

- **introdução**: é feita uma descrição geral da ferramenta e os requisitos mínimos de *software* e *hardware* necessários para a utilizar;
- **como processar dados genéticos**: é explicado como se carregam ficheiros para a área de trabalho, como se definem as opções de processamento, como se processam as distâncias conjuntas e individuais, e como se visualizam os resultados;
- **como usar ficheiros binários**: é explicado como se carregam e processam ficheiros binários;
- **como armazenar dados**: é explicado como se guardam os resultados em ficheiros no disco;
- **descrição dos menus**: é feita uma descrição dos menus e submenus da ferramenta.

Com este documento, espera-se que o utilizador consiga utilizar todas as funcionalidades da ferramenta sem qualquer dificuldade. O manual de utilizador da ferramenta encontra-se no anexo A. A razão deste documento estar escrito em inglês reside no facto da ferramenta também estar em inglês, tornando-a acessível a um maior número de pessoas.

Para além disso, durante a fase de programação foi também criada toda a documentação a partir do *Javadoc*, por forma a documentar a ferramenta desenvolvida. Esta documentação consiste num ficheiro *html* onde se encontram descritas todas as classes existentes no programa bem como as suas funcionalidades e modo de utilização. Este ficheiro é essencial para possíveis desenvolvimentos futuros da ferramenta AGenDA.

Capítulo 4

Resultados

A ferramenta computacional AGenDA, desenvolvida neste trabalho, fornece a possibilidade de criar uma série de estudos diferentes baseados nas distâncias inter-simbólicas, através de uma interface gráfica apelativa e de fácil uso.

Este capítulo pretende mostrar algumas aplicações da ferramenta e os seus respectivos resultados, que podem ser apresentados estatisticamente em gráficos ou então em tabelas com as ocorrências absolutas, permitindo o seu posterior uso.

Para além disso, serão também feitos testes de desempenho com o objectivo de dar a conhecer os tempos médios de execução de genomas completos de algumas espécies.

4.1 Estudos efectuados

Com o objectivo de mostrar exemplos de utilização da ferramenta criada, foram feitos vários estudos que englobam algumas das opções de processamento que ela disponibiliza. Assim, serão processadas as distâncias de um genoma para palavras de diferentes tamanhos e será feito um estudo comparativo entre genomas de diferentes espécies. Serão apresentados alguns resultados desses estudos na forma de tabelas, histogramas e gráficos, assim como algumas conclusões retiradas dos mesmos.

4.1.1 Análise das distâncias do genoma humano completo

Uma possível aplicação desta ferramenta consiste em analisar as distribuições das distâncias de um genoma, para palavras de vários tamanhos. Dado o grande interesse da caracterização do genoma humano, optou-se por fazer um estudo do genoma humano completo para palavras de comprimento um, dois e três. A razão de não se estender a palavras de comprimento superior a três reside no facto da ferramenta criada apenas suportar o processamento de distâncias até tamanho três para genomas de grandes dimensões, como é o caso do genoma humano usado neste estudo.

Foram então calculadas as distribuições das frequências relativas para as distâncias correspondentes aos três tamanhos de palavras processadas. Os histogramas da Figura 4.1 mostram os resultados da distribuição da frequência relativa conjunta e individual das distâncias entre nucleótidos (A, C, G e T). Há que salientar que para palavras de tamanho um é igual considerar o uso de *reading frames* ou não. Por este motivo, apenas se apresenta aqui um dos resultados. Na distribuição da frequência relativa das distâncias entre nucleótidos, os valores

das primeiras distâncias são consideravelmente mais elevados do que as seguintes, decrescendo à medida que a distância aumenta. Significa então que para palavras de comprimento um existe um maior número de distâncias curtas entre nucleótidos e poucas distâncias longas. Seria de esperar este resultado se a sequência de nucleótidos fosse gerada por um processo aleatório i.i.d..

A Figura 4.2 apresenta parte das distribuições das frequências relativas das distâncias entre dinucleótidos com e sem *reading frames*. Para simplificar, são apenas apresentadas as distribuições relativas das palavra AA e CG que retratam os dois tipos de histogramas obtidos. Analisando as palavras de um modo geral, podemos dizer que se obtiveram resultados muito semelhantes aos obtidos para comprimento um, como é o caso da palavra AA que resulta num histograma em que as distâncias mais curtas ocorrem um número de vezes muito superior às distâncias mais longas. No entanto, para a palavra GC e para a CG que é apresentada na Figura 4.2 a distribuição é significativamente diferente, não havendo um decréscimo tão acentuado como se verifica para as outras palavras, verificando-se por vezes um aumento do número de ocorrências de uma distância para a outra. Como se pode verificar, existem algumas diferenças entre as duas abordagens usadas, com e sem *reading frames*, o que é consequência das distâncias serem calculadas de forma diferente, como foi explicado na secção 3.3.2. De facto, quando se considera o uso de *reading frames*, o número de distâncias de tamanho dois é superior relativamente à abordagem em que não se usa *reading frames*.

Na Figura 4.3 são apresentadas algumas distribuições relativas das distâncias entre trinucleótidos sem *reading frames*. Através da amostra de cinco palavras diferentes fornecida nesta figura, podemos constatar que para palavras de tamanho três existem distribuições muito variadas. A distribuição da palavra AAA mantém-se algo semelhante às distribuições descritas anteriormente, para palavras de tamanho um e algumas de tamanho dois. A palavra ATG também se assemelha de certa forma ao padrão obtido para algumas palavras de tamanho dois (CG e GC). No entanto, as palavras CGA, GGC e TCG apresentam características bastante diferentes das descritas até agora, verificando-se picos muito elevados para algumas distâncias em particular. Por exemplo, para a palavra CGA e TCG verifica-se um maior número de ocorrências das distâncias 6, 12, 14 e 20.

A abordagem com *reading frames* apresenta algumas diferenças face à que acabámos de analisar (Figura 4.4). Para a palavra AAA a diferença entre número de ocorrências da distância 1 e das seguintes não é tão acentuada como na abordagem sem *reading frames*, tal como se verificou também para palavras de tamanho dois. A distribuição de distâncias da palavra ATG não difere muito da abordagem anterior, embora se destaque a distância 2 como sendo a que ocorre mais vezes. O mesmo se pode dizer para a palavra GGC, que apresenta um valor muito elevado na distância 2, seguindo-se a distância 13. Há que salientar ainda que tal como se verificou nos resultados anteriores, a frequência relativa da distância 1 da palavra GGC é muito inferior ao habitual. Para as palavras CGA e TCG existe um pico muito superior na distância 10 comparativamente com as outras distâncias.

Através destes histogramas podemos ter uma visão geral do número de ocorrências de cada distância para os diferentes tamanhos de palavras e tentar descobrir padrões que nos ajudem a compreender a estruturar do ADN.

Foram comparados os perfis de erros relativos do genoma humano completo com os das regiões codificantes, de maneira a diferenciar comportamentos em termos da evolução selectiva. Como se pode observar na Figura 4.5, existem claramente diferenças entre os perfis das duas distribuições das distâncias conjuntas. De facto, à região codificante é associado um comportamento oscilatório que não é visível na caso do genoma completo, sendo mais evidente

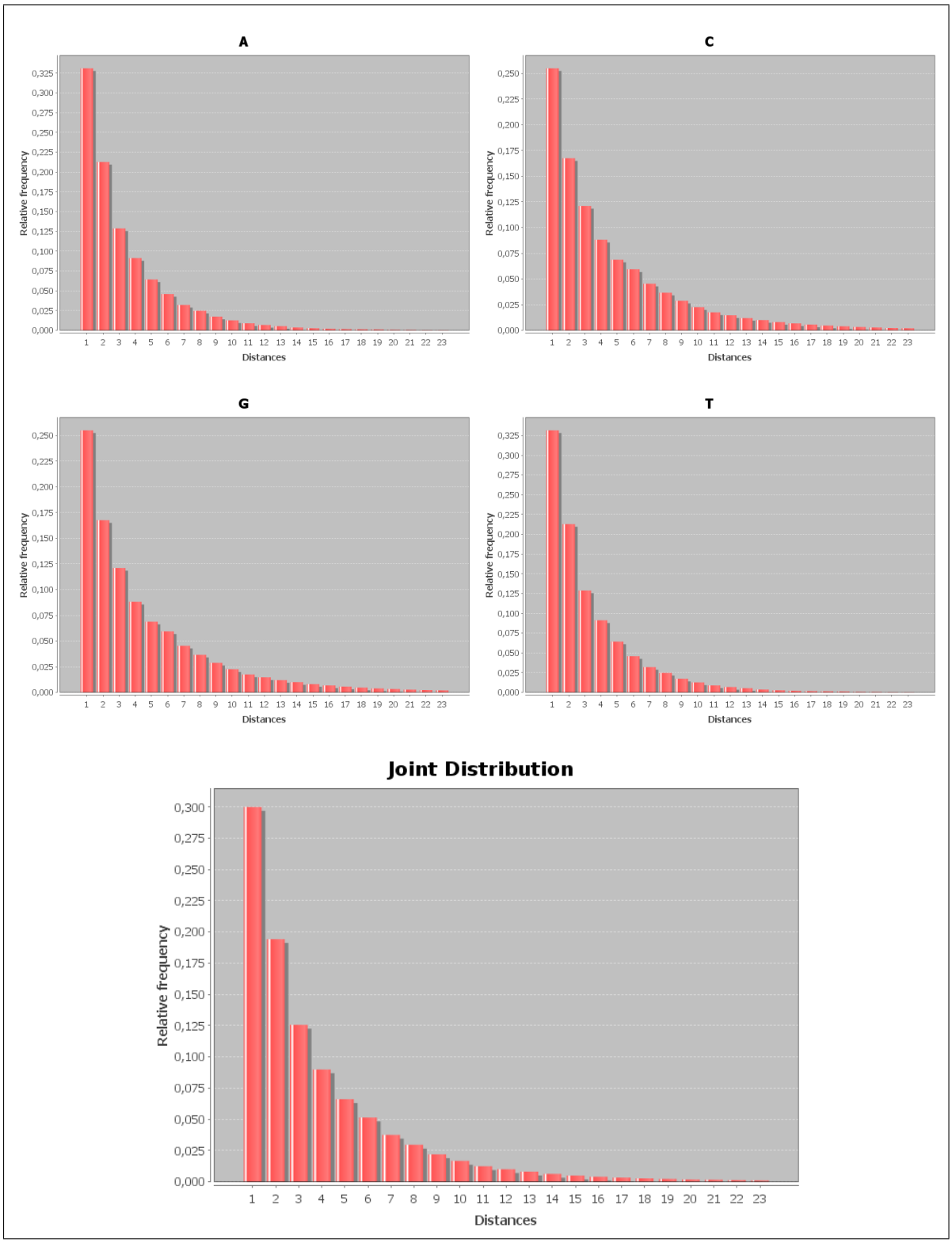


Figura 4.1: Distribuição da frequência relativa das distâncias entre nucleótidos do genoma completo do *Homo sapiens*.

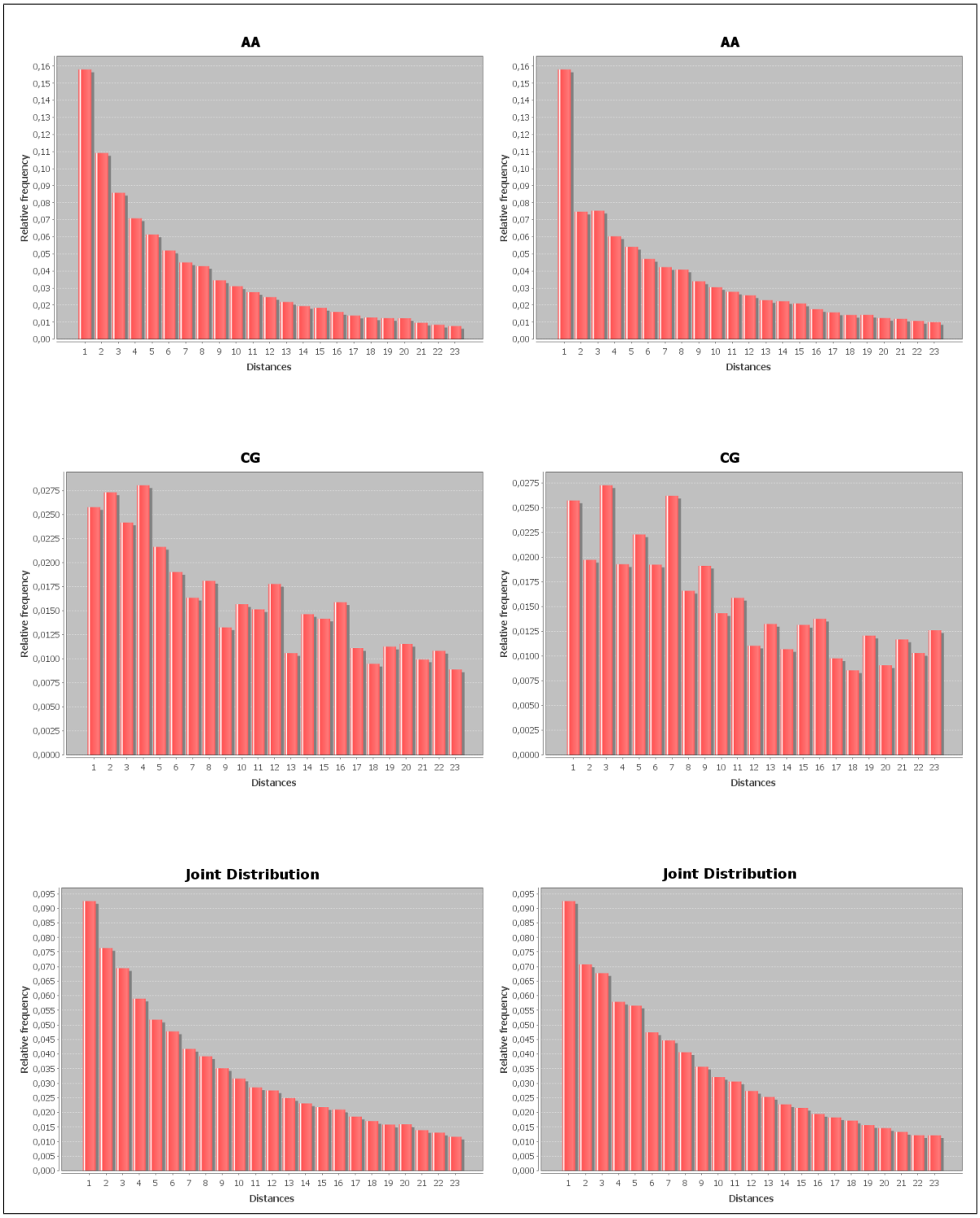


Figura 4.2: Distribuição da frequência relativa das distâncias entre dinucleótidos do genoma completo do *Homo sapiens* com *reading frame* (à esquerda) e sem *reading frame* (à direita). Para facilitar, apenas são apresentados os histogramas da frequência conjunta e das palavras AA e CG.

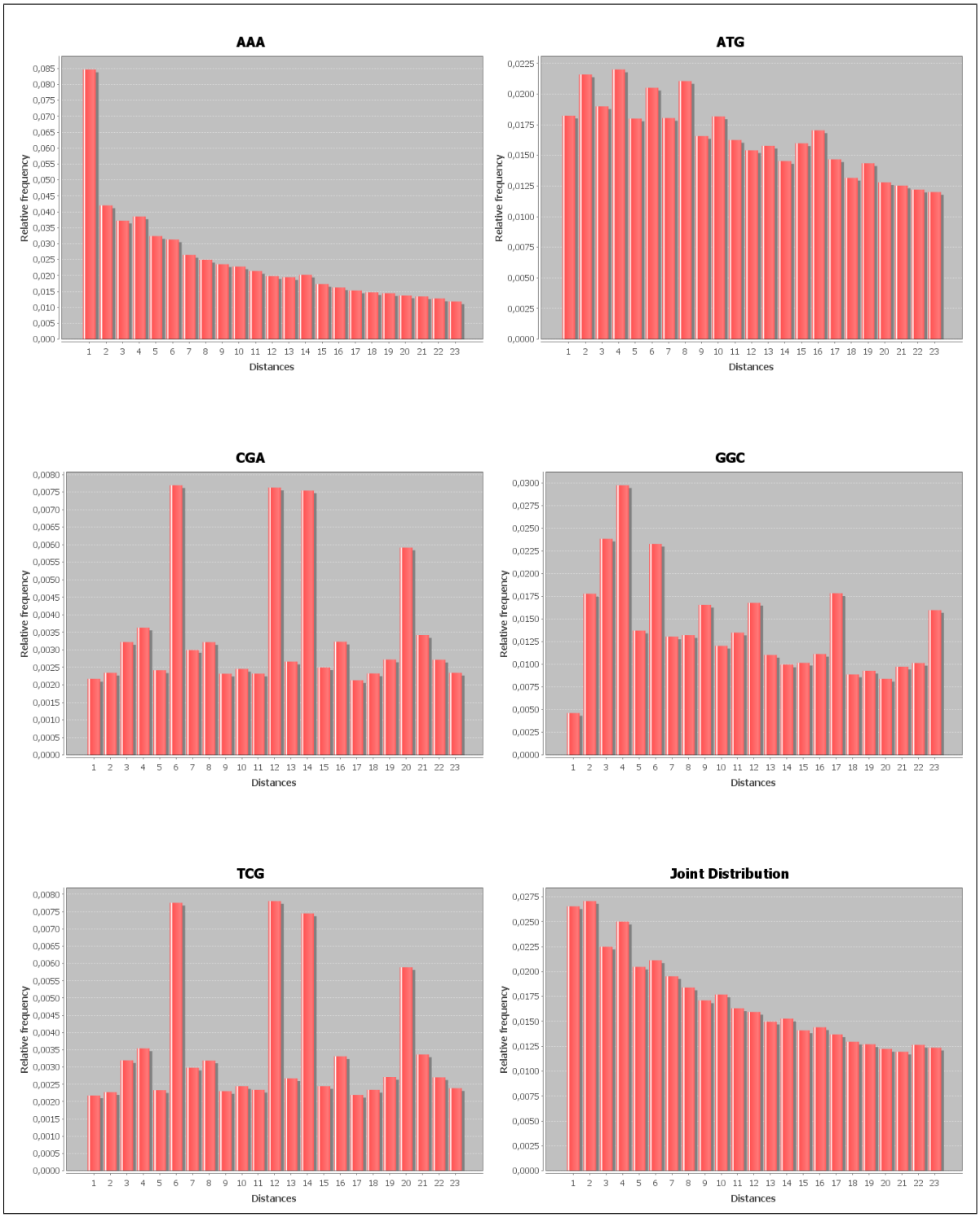


Figura 4.3: Distribuição da frequência relativa das distâncias entre trinucleótidos do genoma completo do *Homo sapiens* sem *reading frame*. Para facilitar, apenas são apresentados histogramas da frequêcia conjunta e de 5 palavras.

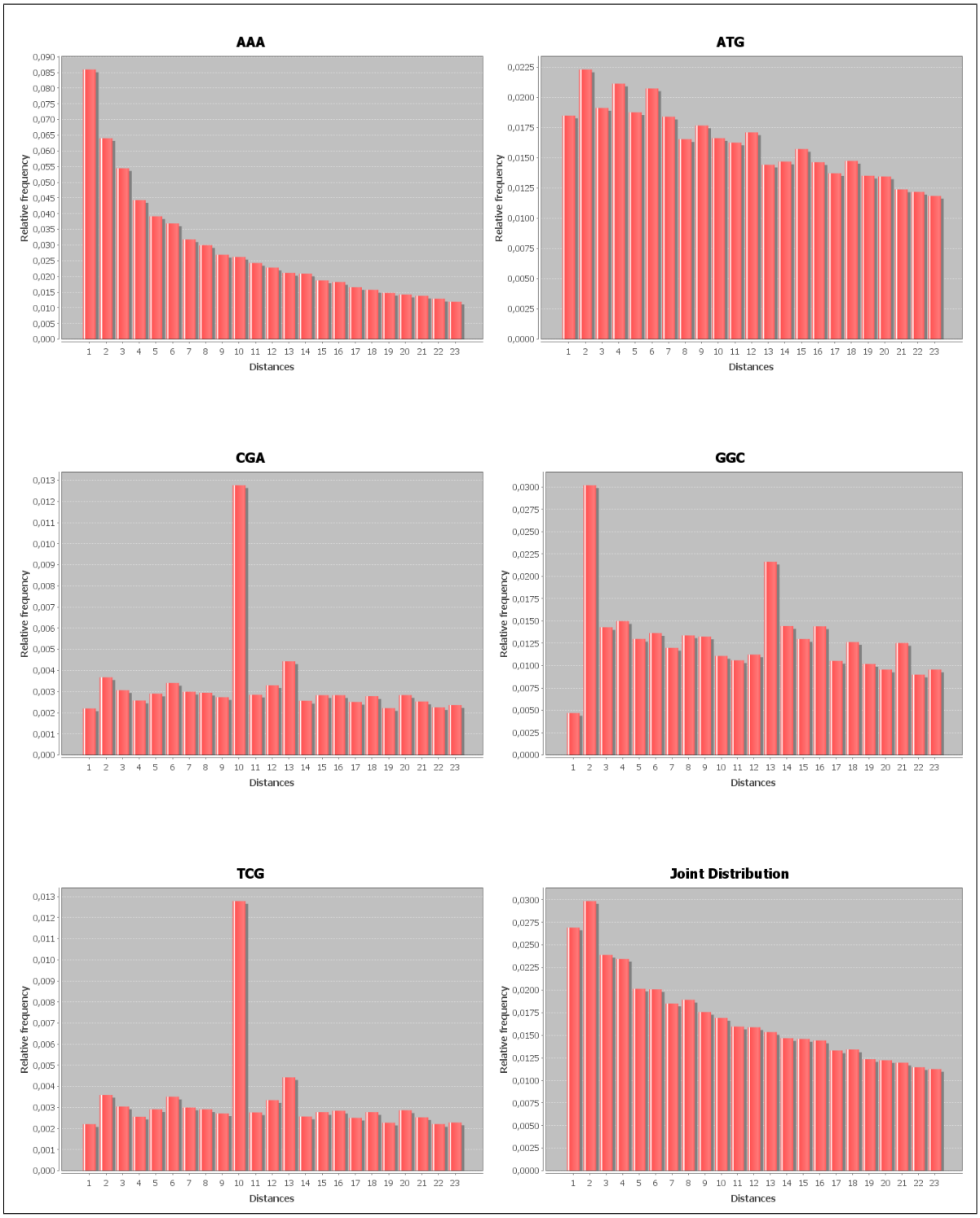


Figura 4.4: Distribuição da frequência relativa das distâncias entre trinucleótidos do genoma completo do *Homo sapiens* com *reading frame*. Para facilitar, apenas são apresentados histogramas da frequêcia conjunta e de 5 palavras.

para palavras de tamanho dois onde se verifica um comportamento constante. No gráfico dos trinucleótidos, a zona codificante não possui um comportamento oscilatório tão evidente, existindo um pico muito elevado para a distância vinte e oito. Na verdade, o genoma humano completo apresenta oscilações mais acentuadas na distribuição das distâncias das palavras de comprimento três que não se verificavam para palavras de comprimento um e dois.

Por fim, obtiveram-se tabelas com a divergência de *Kullback-Leibler* que visam analisar a semelhança entre as distribuições da frequência relativa das distâncias entre todas as palavras do mesmo tamanho. Os resultados da tabela 4.1 confirmam que a distribuição das distâncias entre nucleótidos é semelhante para A e T, assim como para C e G. O que significa não só que no genoma humano completo o número de ocorrências do nucleótido A é semelhante ao número de ocorrências do nucleótido T, assim como o número de ocorrências do nucleótido C é semelhante a número de ocorrências do nucleótido G, mas também que as distribuições das distâncias são muito semelhantes. Na tabela 4.2 são apresentados apenas os resultados da divergência de *Kullback-Leibler* utilizando *reading frames* para palavras de tamanho dois, uma vez que não existem diferenças significativas entre os resultados das duas abordagens. Observam-se então semelhanças entre as distribuições das distâncias e o seu complemento invertido (AA-TT, AC-GT, AG-CT, CA-TG, CC-GG, GA-TC). Devido às suas grandes dimensões, a tabela de *Kullback-Leibler* para palavras de comprimento três não é apresentada neste trabalho, mas continuam a verificar-se semelhanças relativamente às frequências das palavras e do seu complemento invertido.

Tabela 4.1: Divergência de *Kullback-Leibler* da distribuição das distâncias entre nucleótidos do genoma completo do *Homo sapiens*.

	A	**C**	**G**	**T**
A	0.000	0.059	0.059	0.000
C	0.076	0.000	0.000	0.076
G	0.075	0.000	0.000	0.076
T	0.000	0.059	0.059	0.000

Tabela 4.2: Divergência de *Kullback-Leibler* da distribuição das distâncias entre dinucleótidos do genoma completo do *Homo sapiens*, com *reading frame*.

	AA	**AC**	**AG**	**AT**	**CA**	**CC**	**CG**	**CT**	**GA**	**GC**	**GG**	**GT**	**TA**	**TC**	**TG**	**TT**
AA	0.000	0.187	0.073	0.042	0.077	0.137	0.476	0.073	0.101	0.409	0.137	0.186	0.060	0.101	0.076	0.000
AC	0.215	0.000	0.054	0.096	0.067	0.026	0.125	0.054	0.017	0.088	0.026	0.000	0.052	0.017	0.068	0.217
AG	0.073	0.049	0.000	0.010	0.004	0.026	0.255	0.000	0.015	0.137	0.026	0.048	0.006	0.015	0.004	0.074
AT	0.041	0.084	0.010	0.000	0.008	0.046	0.312	0.010	0.032	0.186	0.046	0.083	0.009	0.032	0.008	0.041
CA	0.073	0.057	0.004	0.008	0.000	0.036	0.272	0.004	0.022	0.140	0.036	0.057	0.010	0.022	0.000	0.074
CC	0.151	0.028	0.031	0.055	0.045	0.000	0.155	0.031	0.016	0.072	0.000	0.028	0.024	0.016	0.045	0.152
CG	0.696	0.168	0.401	0.484	0.456	0.191	0.000	0.402	0.254	0.130	0.191	0.170	0.333	0.255	0.458	0.700
CT	0.073	0.049	0.000	0.010	0.004	0.026	0.255	0.000	0.015	0.137	0.026	0.049	0.006	0.015	0.004	0.074
GA	0.114	0.017	0.017	0.036	0.025	0.016	0.188	0.017	0.000	0.126	0.016	0.016	0.013	0.000	0.025	0.115
GC	0.338	0.058	0.121	0.172	0.136	0.055	0.111	0.121	0.083	0.000	0.055	0.058	0.117	0.083	0.137	0.340
GG	0.151	0.028	0.031	0.055	0.045	0.000	0.155	0.031	0.016	0.072	0.000	0.028	0.024	0.016	0.045	0.152
GT	0.214	0.000	0.053	0.095	0.066	0.026	0.126	0.053	0.016	0.089	0.026	0.000	0.051	0.016	0.067	0.216
TA	0.063	0.050	0.006	0.010	0.011	0.024	0.249	0.006	0.013	0.152	0.023	0.049	0.000	0.013	0.011	0.064
TC	0.114	0.017	0.017	0.036	0.025	0.016	0.188	0.017	0.000	0.126	0.016	0.016	0.013	0.000	0.025	0.115
TG	0.073	0.058	0.004	0.008	0.000	0.036	0.273	0.004	0.022	0.140	0.036	0.057	0.010	0.022	0.000	0.074
TT	0.000	0.189	0.074	0.043	0.078	0.138	0.477	0.074	0.102	0.410	0.138	0.188	0.061	0.102	0.077	0.000

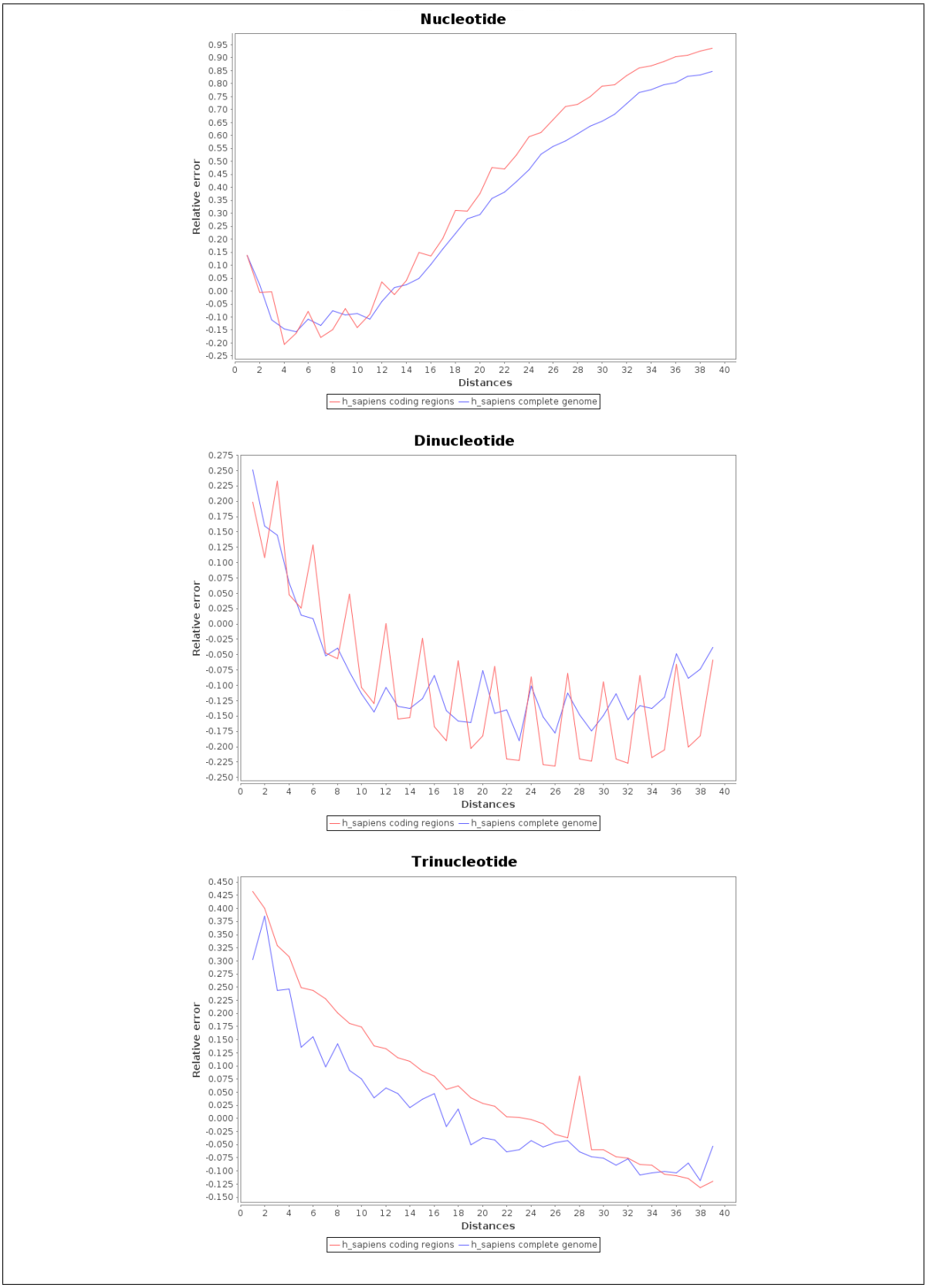


Figura 4.5: Erro relativo das distâncias entre nucleótidos, dinucleótidos e trinucleótidos do genoma completo e da parte codificante do *Homo sapiens*, sem *reading frame*.

4.1.2 Análise comparativa de genomas de diferentes espécies

Outro exemplo de utilização da ferramenta consiste em comparar genomas de diferentes espécies. Para isso, foram calculadas as distribuições das distâncias entre nucleótidos das espécies seguintes:

- *Homo sapiens*;
- *Macaca mulatta*;
- *Mus musculus*;
- *Pan troglodytes*;
- *Escherichia coli*;
- *Rattus norvegicus*;
- *Streptococcus mutans*;
- *Streptococcus pneumoniae*.

Como podemos verificar, os resultados obtidos no gráfico da Figura 4.6, confirmam a existência de uma assinatura genética que identifica cada espécie.

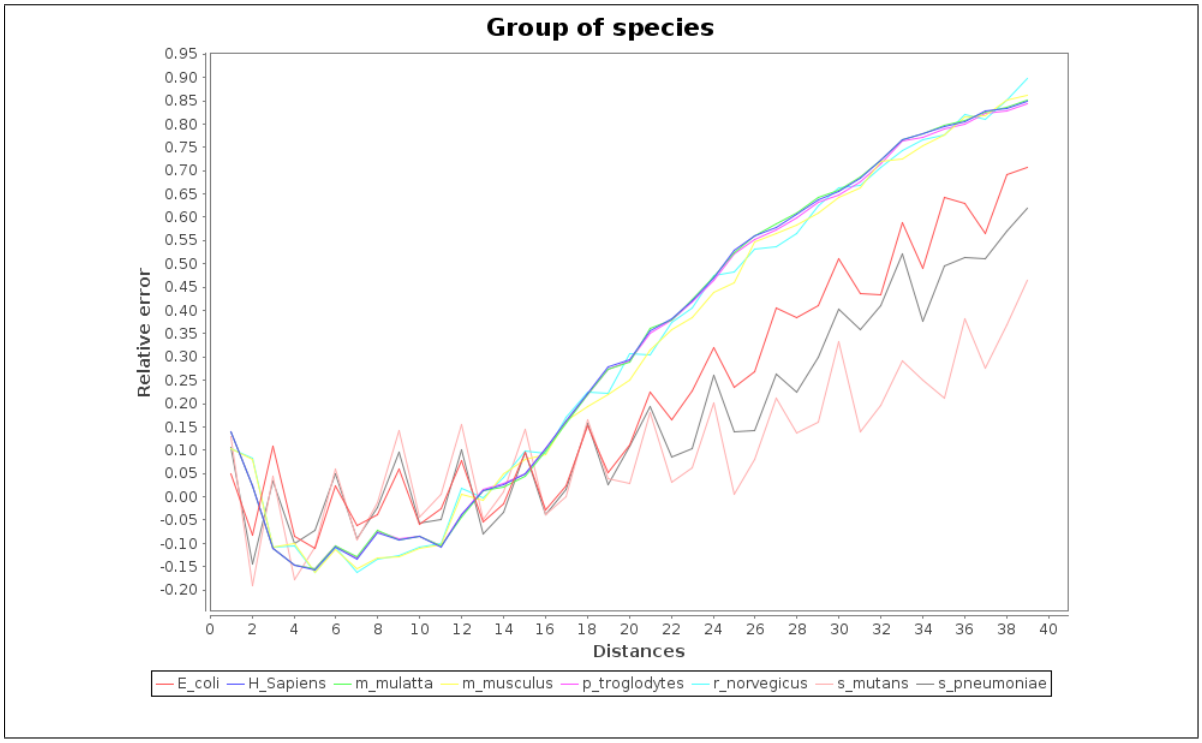


Figura 4.6: Erro relativo do genoma completo de um grupo de espécies para palavras de tamanho 1 e distância máxima de 100, sem *reading frame*. As espécies usadas foram: *Homo sapiens, Macaca mulatta, Mus musculus, Pan troglodytes, Escherichia coli, Rattus norvegicus, Streptococcus mutans and Streptococcus pneumoniae.*

De facto, os perfis dos erros relativos apresentam linhas muito semelhantes para os mamíferos, como é o caso do *Homo sapiens, Macaca mulatta, Mus musculus, Pan troglodytes* e *Rattus norvegicus*. As bactérias, por sua vez, possuem um erro relativo muito diferente quando comparado com o dos mamíferos, mas bastante semelhante entre elas. O que é de esperar uma vez que o genoma das bactérias contém praticamente só parte codificante.

4.2 Testes de Desempenho

Uma vez que este trabalho lida com ficheiros de grandes dimensões é importante fazer vários testes utilizando ficheiros de várias dimensões, assim como as diferentes opções possíveis de processamento. Desta forma, serão efectuados cálculos das distâncias para vários genomas completos. Todos os resultados apresentados nesta secção foram processados numa máquina com as seguintes características:

- Tipo de processador: Intel Core 2 Duo T6400;
- Velocidade do processador: 2.2 GHz;
- Memória RAM: 4GB DDR2;
- Sistema operativo: *Windows 7 Ultimate Edition*

Foram então processadas as distâncias de palavras de tamanho um dois e três para diversos genomas, utilizando as duas abordagens disponibilizadas. Os resultados estão apresentados em duas tabelas distintas, uma para a abordagem com *reading frame* e outra para a abordagem sem *reading frame*. Os tempos médios de processamento foram calculados a partir da média ponderada de 10 execuções em ambientes semelhantes.

Tabela 4.3: Tempos de execução do processamento de genomas completos sem *reading frame*.

Espécie	**Tamanho dos ficheiros**	**Tempos de execução**		
		Tamanho 1	Tamanho 2	Tamanho 3
Homo sapiens	2.7 GB	20.3 min.	31.3 min.	40.7 min.
Pan troglodytes	3.0 GB	22.4 min.	30.1 min	38.2 min.
Macaca mulatta	2.0 GB	15.0 min.	21.2 min.	23.9 min
Mus musculus	2.6 GB	20.9 min.	28.3 min.	37.0 min.
Equus caballus	2.0 GB	17.6 min	23.5 min.	30.5 min.
Rattus norvegicus	2.7 GB	21.9 min	27.2 min.	33.9 min.
Gallus gallus	1.1 GB	8.5 min.	11.3 min.	14.1 min.
Vitis vinifera	471 MB	3.8 min	5.3 min.	6.2 min.
Escherichia coli	4.5 MB	2.9 seg.	3.5 seg.	6.69 seg.
Streptococcus mutans	2.0 MB	1.04 seg.	1.57 seg.	3.87 seg.
Streptococcus pneumoniae	2.1 MB	1.67 seg.	1.87 seg.	5.14 seg.

Tabela 4.4: Tempos de execução do processamento de genomas completos com *reading frame*.

Espécie	**Tamanho dos ficheiros**	**Tempos de execução**		
		Tamanho 1	Tamanho 2	Tamanho 3
Homo sapiens	2.7 GB	19.8 min.	26.1 min	38.9 min
Pan troglodytes	3.0 GB	19.0 min	25.7 min	37.2 min.
Macaca mulatta	2.0 GB	12.7 min.	17.9 min.	21.0 min.
Mus musculus	2.6 GB	17.8 min.	23.6 min.	35.8 min.
Equus caballus	2.0 GB	15.0 min.	19.9 min.	26.8 min.
Rattus norvegicus	2.7 GB	17.7 min.	22.1 min.	28.9 min.
Gallus gallus	1.1 GB	7.2 min.	9.1 min.	13.2 min.
Vitis vinifera	471 MB	3.1 min.	4.2 min.	5.1 min.
Escherichia coli	4.5 MB	2.01 seg.	2.84 seg.	6.99 seg.
Streptococcus mutans	2.0 MB	0.91 seg.	1.20 seg.	4.36 seg.
Streptococcus pneumoniae	2.1 MB	0.98 seg.	1.38 seg.	4.86 seg.

Capítulo 5

Conclusão e trabalho futuro

5.1 Conclusão

A sequenciação do primeiro genoma despoletou a necessidade de descobrir mecanismos que permitissem estudar e compreender a sua estrutura. Com o melhoramento das técnicas de sequenciação, foram sendo geradas grandes quantidades de dados genéticos o que fez com que surgissem novos desafios, nomeadamente a criação de ferramentas de suporte para analisar toda essa informação.

Diferentes tipos de estudo têm sido realizados com o objectivo de descobrir novos padrões genéticos e tentar compreender a relação entre as espécies. Com este trabalho pretendeu-se dar um contributo nesta área através do desenvolvimento de uma ferramenta computacional direccionada ao estudo do mapeamento das distâncias.

O mapeamento das distâncias inter-simbólicas tinha demonstrado ser de grande utilidade no que diz respeito à comparação de sequências de ADN e procura de semelhanças entre diferentes genomas.

A ferramenta AGenDA desenvolvida neste trabalho permite então estudar a distribuição das distâncias de oligonucleótidos de diferentes tamanhos, fornecendo uma interface gráfica intuitiva que facilita a interacção com o utilizador. Esta interface possibilita não só o processamento integral de vários ficheiros, como também a selecção visual da zona a processar de uma forma simples e fácil de usar.

O processamento das distâncias foi feito utilizando duas abordagens distintas. A primeira calcula as distâncias entre oligonucleótidos usando *reading frames* que consistem em blocos do tamanho dos oligonucleótidos estipulados *à priori*. O número de *frames* existente é igual ao tamanho dos oligonucleótidos, e é calculada uma sequência de distâncias para cada *frame*. Assim, é criada uma primeira sequência de distâncias relativa à primeira *reading frame* que começa na posição zero da sequência, de seguida uma segunda sequência de distâncias que começa uma posição à frente da primeira e assim sucessivamente até ser atingido o número total de *frames*. A outra abordagem não considera nenhuma *reading frame*, procura apenas a ocorrência de palavras equivalentes e calcula a distância entre elas.

De maneira a facilitar a interpretação dos resultados, foram utilizados vários métodos quantitativos que permitem uma visualização gráfica dos mesmos. Assim, é possível ter uma perspectiva geral da distribuição das distâncias entre os oligonucleótidos a partir dos histogramas criados pela ferramenta. O uso da divergência de *Kullback-Leibler* permite analisar as semelhanças das distribuições das distâncias entre todas as palavras do mesmo tamanho

e visualizar estes resultados através de tabelas. Para além disso, a ferramenta proporciona a possibilidade de estudos comparativos entre diferentes genomas, criando para esse efeito gráficos que usam o erro relativo entre a frequência relativa das distâncias observadas e a probabilidade estimada pela distribuição de referência.

Os estudos realizados com o auxílio desta ferramenta permitiram constatar a existência de padrões genéticos, realçando a relação evolutiva das espécies. Tendo sido verificadas semelhanças entre os mamíferos, como foi o caso do *Homo sapiens, Macaca mulatta, Mus musculus, Pan troglodytes* e *Rattus norvegicus*, e também entre as bactérias como a *Escherichia coli*, *Streptococcus mutans* e *Streptococcus pneumoniae*. Havendo no entanto grandes diferenças entre os padrões dos mamíferos e das bactérias.

Podemos então concluir que o desenvolvimento da ferramenta AGenDA contribuiu de forma significativa para o estudo do mapeamento de distâncias.

5.2 Perspectivas de trabalho futuro

Como pudemos verificar, o desenvolvimento da ferramenta AGenDA contribuiu de forma positiva para o estudo do mapeamento de distâncias. Embora os objectivos principais deste trabalho tenham sido atingidos, ele não é uma versão final, mas sim a base para a desenvolvimento de uma ferramenta mais robusta e eficiente.

Assim, as perspectivas futuras deste trabalho englobam essencialmente o refinamento de alguns pontos e também a inclusão de novas funcionalidades, por forma a tornar a ferramenta ainda mais usável.

No que diz respeito ao processamento do mapeamento de distâncias, há dois aspectos importantes que poderiam melhorar o desempenho do programa que consistem na eficiência e na escalabilidade do algoritmo usado. Seria então vantajoso aperfeiçoar o algoritmo do cálculo das distâncias de maneira a fazer o processamento de uma forma mais rápida e que fosse possível aplicá-lo a distâncias de palavras de tamanhos superiores, sem grandes restrições de memória.

Relativamente à interface gráfica, algumas melhorias poderiam ser feitas a nível de usabilidade. Nomeadamente, testar a ferramenta num ambiente real e tentar entender quais as dificuldades sentidas pelos utilizadores durante a sua utilização.

Poderia também ser interessante incluir outros métodos de análise das sequências de distâncias para além dos que estão disponíveis nesta ferramenta. Embora o objectivo base seja criar dados com informação relativa ao mapeamento de distâncias e estes poderem vir a ser usados mais tarde por ferramentas de análise de dados, poderá ser satisfatório para o utilizador evitar este último passo e obter toda a informação que necessita no mesmo programa.

Por fim, existem uma série de novas funcionalidades que poderiam vir a ser adicionadas à ferramenta, proporcionando uma melhor experiência ao utilizador. Como a possibilidade de agendar diferentes tipos de estudo em simultâneo, fazer o processamento através da linha de comandos e criar a possibilidade de deixar o processamento correr em *background*, permitindo que o utilizador continue a utilizar a sua máquina.

Apêndice A

Manual de utilizacão da ferramenta

University of Aveiro

Department of Electronics, Telecommunications and Informatics

AGenDA - A Genome Distance Analyser User Manual

Susana Vasconcelos

OCTOBER 26, 2011

Contents

A.1

Introduction

A.1.1 Overview

AGenDA is a Java software tool that analyzes the oligonucleotide distances of genome sequences for various word lengths. It computes the distance sequences, distance distribution and genomic profiles based on distance frequency. This tool is able to perform comparative analysis between different genomic sequences (e.g. comparison between species).

A.1.2 System Requirements

Linux (Ubuntu 11.04 or above), Macintosh OS X (10.4, power-book intel) and Windows (2000, XP, Vista or 7); Processor - 800 MHz Intel Pentium III or higher; Memory - 1 GB of RAM (2 GB or more recommended); Disk space available - 100 MB at least, Minimum resolution - 800 x 600 (1024 x 768 recommended); Java Virtual Machine (JRE 6.x)

A.2

How to Process genome Data

AGenDA alows to process genome DNA sequence files in FASTA format, which includes the ".fna", "ffn" and "fa" extensions. These files can be downloaded from the international data banks (ftp://ftp.ncbi.nih.gov/genbank/genomes/).

A.2.1 Uploading FASTA files

1. Go to "Run Fasta File" tab, on the left side and click on the plus (+) icon (Figure A.2.1).
2. Select the folder or the file you want to process. Note that only files in FASTA format will be uploaded to your workspace.
3. Confirm or change the name group for the selected sequence files and press "ok".

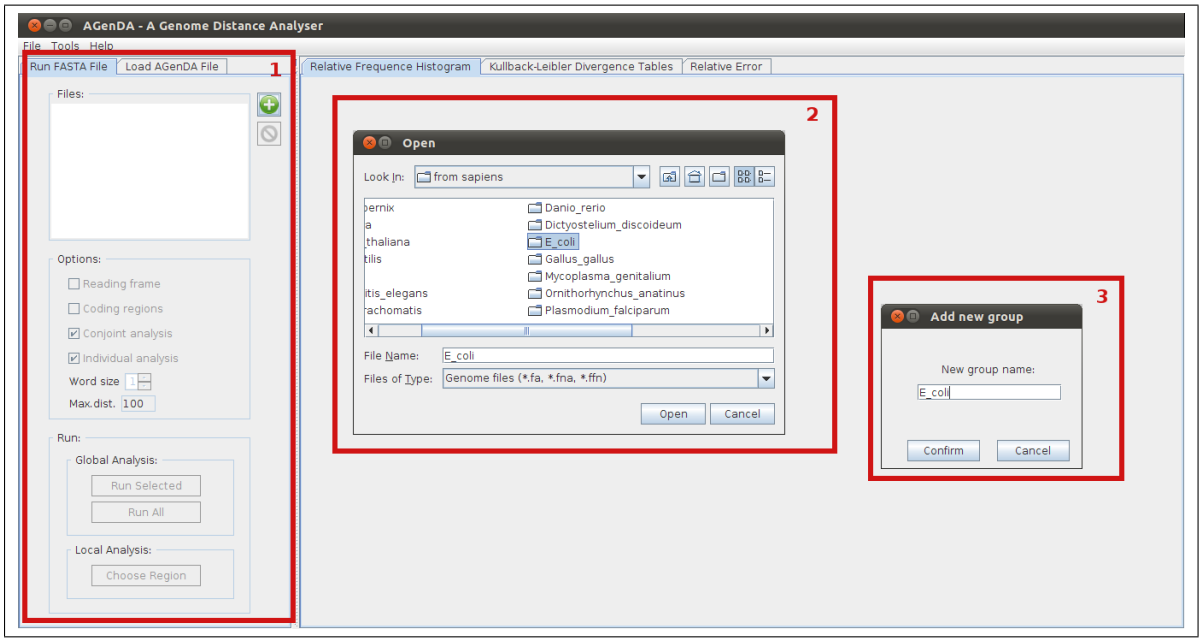
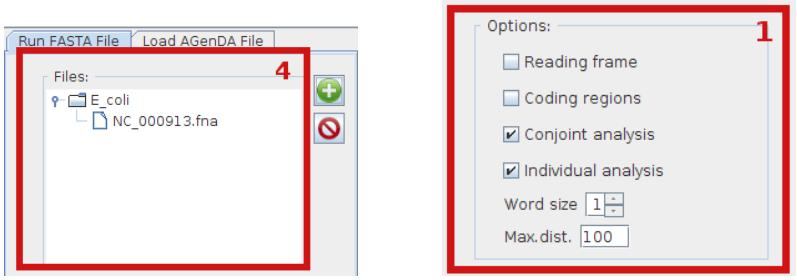


Figure A.2.1: Running FASTA file.

4. Now you have the selected files in your workspace (Figure A.2.2a)



(a) (b)

Figure A.2.2: Selected files in workspace (a) and processing option (b)

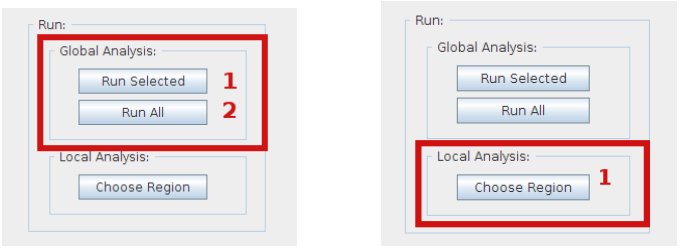
A.2.2 Choosing processing options

1. Check the boxes to choose the type of analysis (Figure A.2.2b):
   - Reading Frame: it uses the reading frame approach;
   - Coding Regions: it considers the coding regions of a file;
   - Conjoint Analysis: it only calculates the total distance distribution of every word;
   - Individual Analysis: it calculates the distance distribution of each word.
2. Define the word size you want to process.
3. Define the maximum distance to consider for the analysis.

A.2.3 Processing distances

A.2.3.1 Processing Total Distances

1. Press the Run Selected button in case you want to only process a selected file. Note: be sure a file is selected, otherwise it will appear a warning window (Figure A.2.3a).
2. Press the Run All button in case you want to process all the files.

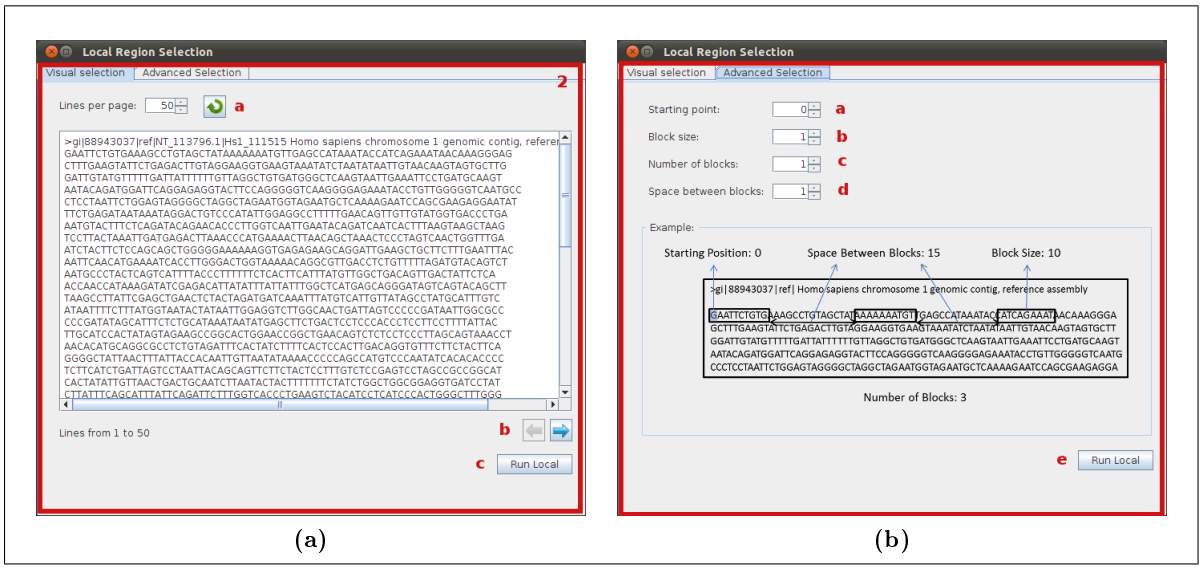


(a) (b)

Figure A.2.3: Conjoint and Individual analysis.

A.2.3.2 Processing Local Distances

1. Press the Choose Region button (Figure A.2.3b) and a window will appear.
2. If you want to select the individual region visually, go to Visual Selection tab (Figure A.2.4a):
   (a) You can change the number of lines that will appear on the box by changing the lines per page spinner and then pressing the refresh button;
   (b) You can move forward and backward through the FASTA file by pressing the rigth and left arrow respectively;
   (c) Select the local region using the mouse and then press Run Local button. Note: if no region is selected a warning window will appear.
3. If you want to select various non continuous local regions at the same time go to Advanced Selection (Figure A.2.4b):
   (a) Set the starting point;
   (b) Set the block size;
   (c) Set the number of block to be processed;
   (d) Set the space between the blocks;
   (e) Press Run Local button.



(a) (b)

Figure A.2.4: Local Region Selection window: Visual Selecion (a) and Advanced Selection (b)

A.2.4 Visualizing Data

A.2.4.1 Relative Frequency Histogram

1. The first histogram to be shown is the Conjoint Histogram (Figure A.2.5).

2. You can change the name of the genome you want to see, using the "Choose genome" spinner. Note that a genome is considered to be a group of files inside a folder in the workspace.

3. You can see the local histograms by clicking on the oligonucleotide combo box (Figure A.2.6).

4. To move forwards and backwards on the oligonucleotide histograms use the next and previous buttons.

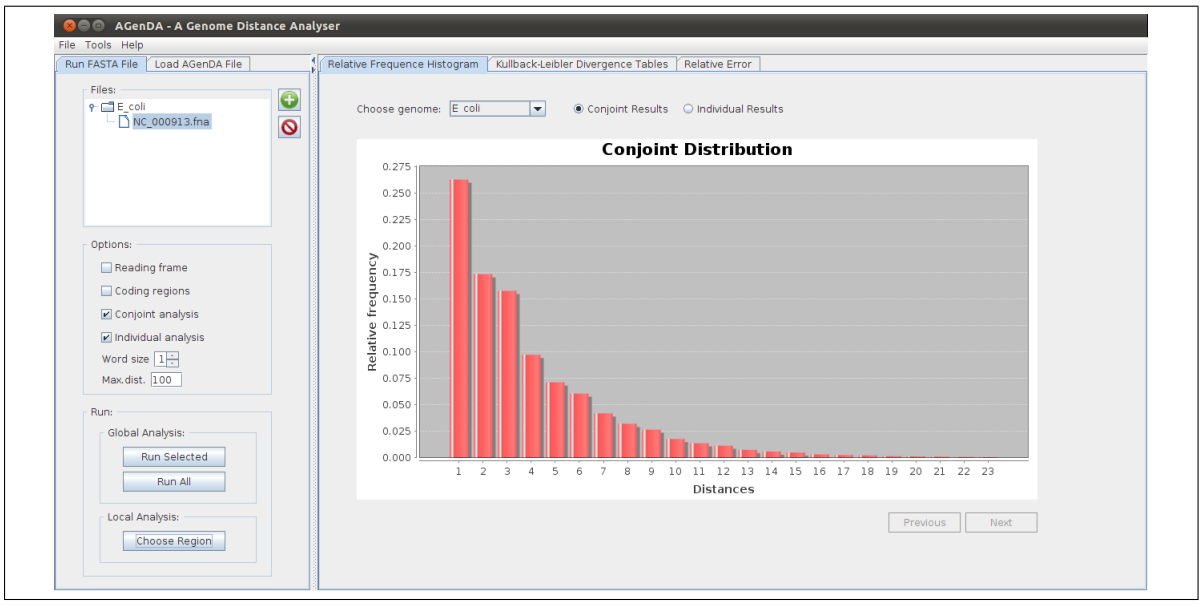


Figure A.2.5: Histogram tab: Conjoint Histogram.

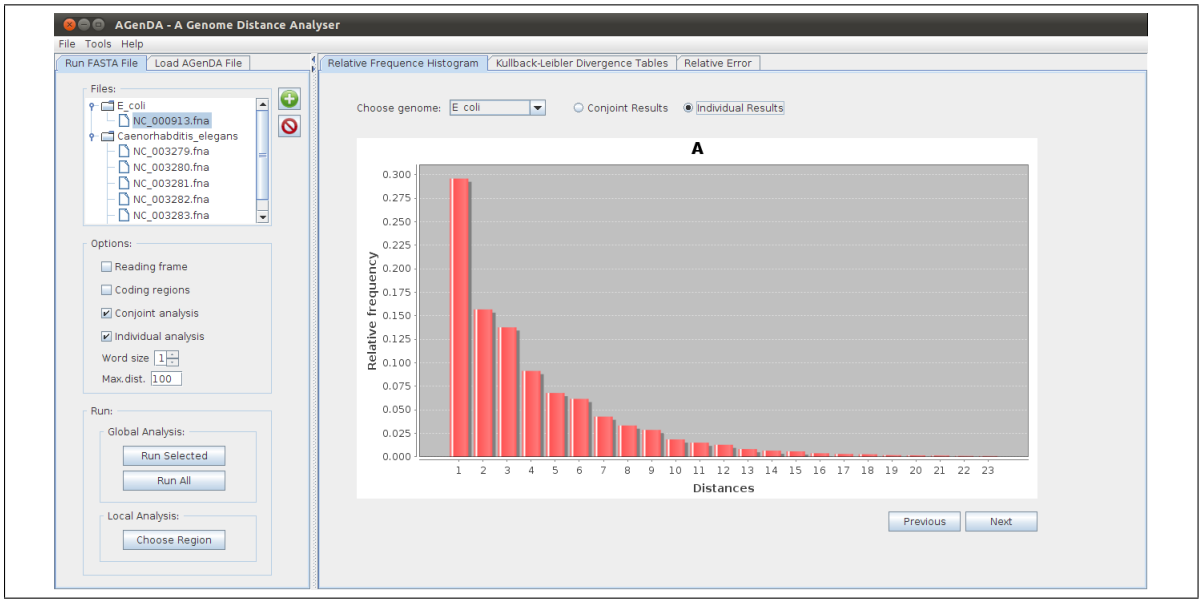
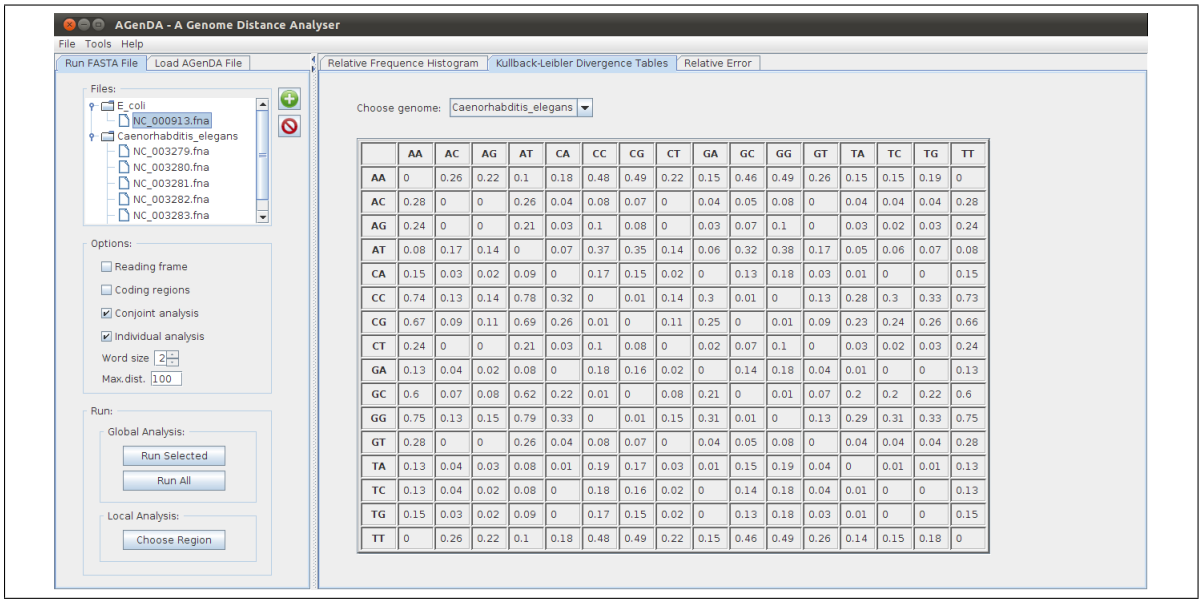


Figure A.2.6: Histogram tab: Individual Histogram.

A.2.4.2 Kullback-Leibler Divergence Tables

1. The table shows the Kullback-Leibler Divergence of the oligonucleotides in the whole genome (Figure A.2.7).

2. You can change the name of the genome you want to see, using the "Choose genome" spinner. Note that a genome is considered to be a group of files inside a folder in the workspace.



	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
AA	0	0.26	0.22	0.1	0.18	0.48	0.49	0.22	0.15	0.46	0.49	0.26	0.15	0.15	0.19	0
AC	0.28	0	0	0.26	0.04	0.08	0.07	0	0.04	0.05	0.08	0	0.04	0.04	0.04	0.28
AG	0.24	0	0	0.21	0.03	0.1	0.08	0	0.03	0.07	0.1	0	0.03	0.02	0.03	0.24
AT	0.08	0.17	0.14	0	0.07	0.37	0.35	0.14	0.06	0.32	0.38	0.17	0.05	0.06	0.07	0.08
CA	0.15	0.03	0.02	0.09	0	0.17	0.15	0.02	0	0.13	0.18	0.03	0.01	0	0	0.15
CC	0.74	0.13	0.14	0.78	0.32	0	0.01	0.14	0.3	0.01	0	0.13	0.28	0.3	0.33	0.73
CG	0.67	0.09	0.11	0.69	0.26	0.01	0	0.11	0.25	0	0.01	0.09	0.23	0.24	0.26	0.66
CT	0.24	0	0	0.21	0.03	0.1	0.08	0	0.02	0.07	0.1	0	0.03	0.02	0.03	0.24
GA	0.13	0.04	0.02	0.08	0	0.18	0.16	0.02	0	0.14	0.18	0.04	0.01	0	0	0.13
GC	0.6	0.07	0.08	0.62	0.22	0.01	0	0.08	0.21	0	0.01	0.07	0.2	0.2	0.22	0.6
GG	0.75	0.13	0.15	0.79	0.33	0	0.01	0.15	0.31	0.01	0	0.13	0.29	0.31	0.33	0.75
GT	0.28	0	0	0.26	0.04	0.08	0.07	0	0.04	0.05	0.08	0	0.04	0.04	0.04	0.28
TA	0.13	0.04	0.03	0.08	0.01	0.19	0.17	0.03	0.01	0.15	0.19	0.04	0	0.01	0.01	0.13
TC	0.13	0.04	0.02	0.08	0	0.18	0.16	0.02	0	0.14	0.18	0.04	0.01	0	0	0.13
TG	0.15	0.03	0.02	0.09	0	0.17	0.15	0.02	0	0.13	0.18	0.03	0.01	0	0	0.15
TT	0	0.26	0.22	0.1	0.18	0.48	0.49	0.22	0.15	0.46	0.49	0.26	0.14	0.15	0.18	0

Figure A.2.7: Kullback-Leibler Divergence Tables.

A.2.4.3 Relative Error

1. The chart shows the relative error of the whole genome (Figure A.2.8).

2. You can change the name of the genome you want to see, using the "Choose genome" spinner. Note that a genome is considered to be a group of files inside a folder in the workspace.

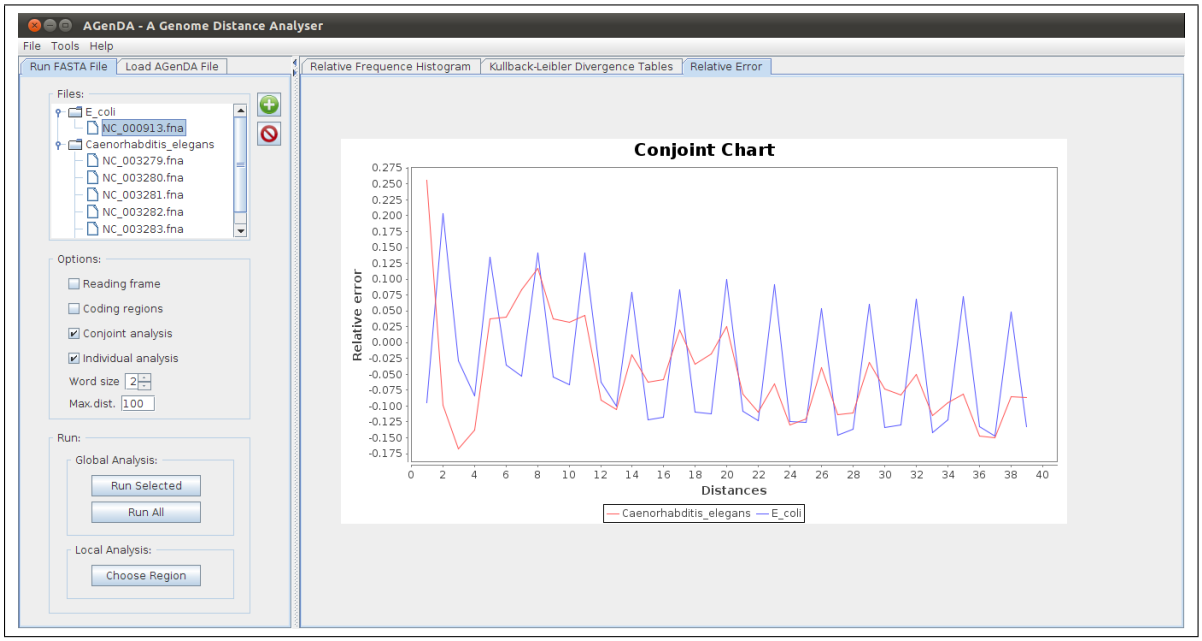


Figure A.2.8: Relative Error Tab.

A.3

How to use Binary Files

AGenDA alows to upload a binary file that has been processed previously and includes all the information about the processed files and its distance distribution results. Using this file you are able to create charts, histograms and tables of a large number of files very quickly.

A.3.1 Uploading binary file

1. Go to "Load AGenDA File" tab, on the left side and click on the plus (+) icon (Figure A.3.1).

2. Select the AGenDA file you want to process. Note that only files in AGenDA format will be uploaded to your workspace.

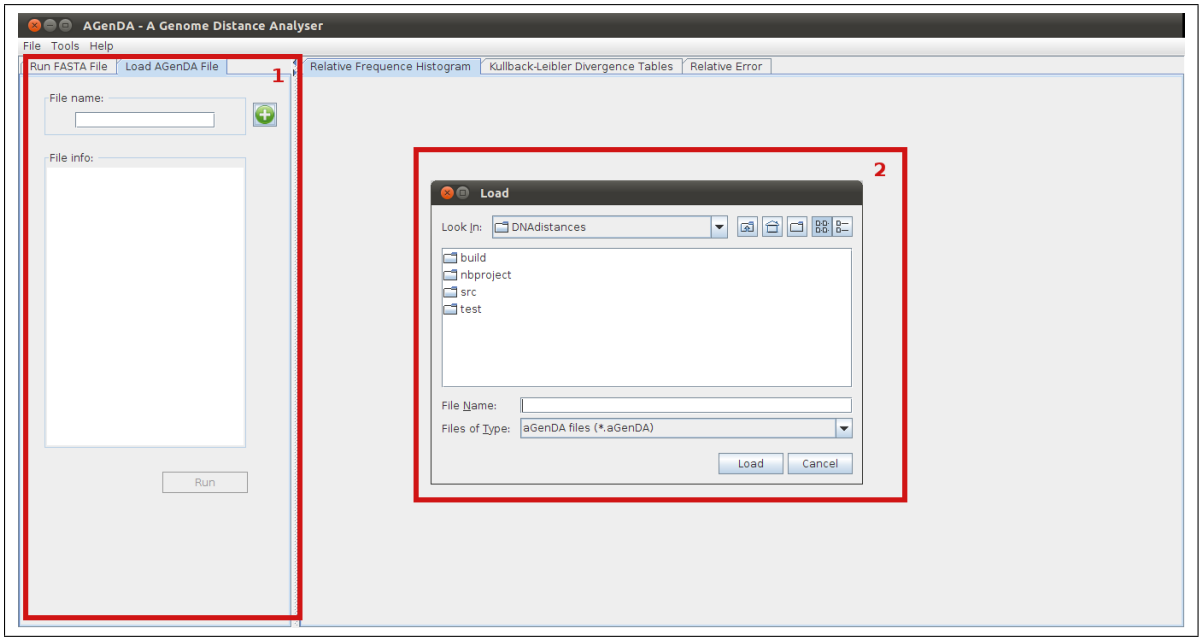


Figure A.3.1: Uploading binary file.

A.3.2 Processing binary file

1. Inside the file info box you can see all the information available about the genome(s) included in the loaded AGenDA file (Figure A.3.2).

2. Press the "Run" button and you will get all the graphics and tables described on section A.2.4.

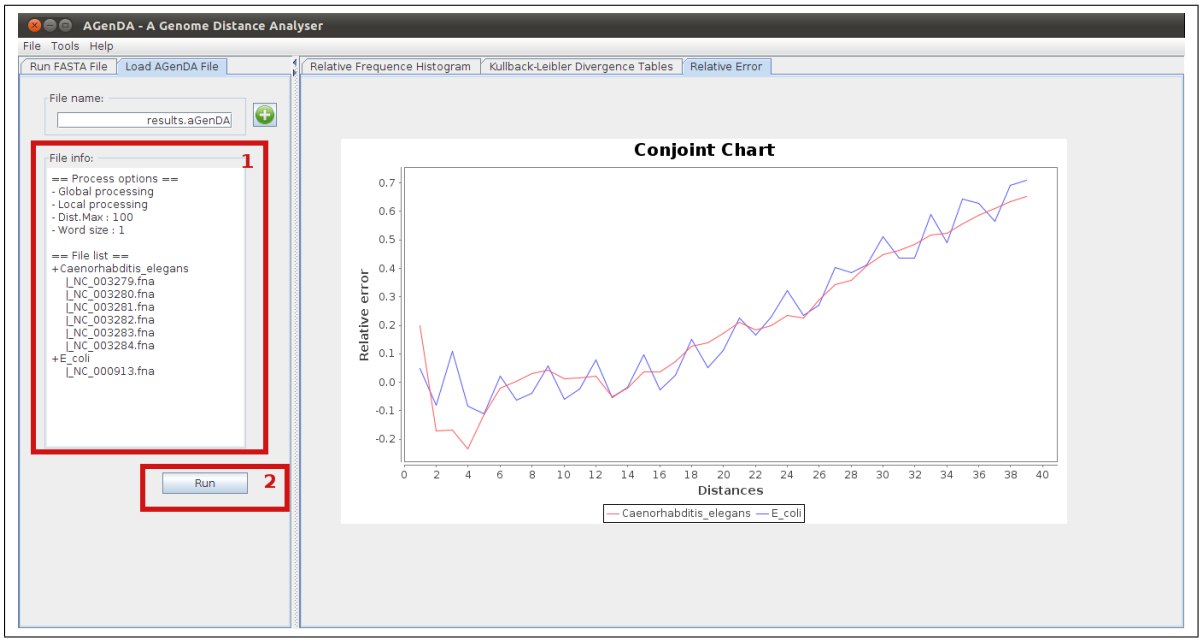


Figure A.3.2: Menu options: file menu (a), tools menu (b) and help menu (c).

A.4

How to save Data

AGenDA alows to save all the obtained results. That includes the graphic visualizations described on section A.2.4 and also a set of tables with distance sequence data used to create the graphs. It also permits to save a binary file in AGenDA format that includes all the data needed to crate the graphic visualizations.

A.4.1 Saving Data

1. In order to save data go to the "Tools" menu and then "Save Results" submenu.
2. The save options window is divided in four tabs:
   (a) Histograms: include Conjoint and Individual Relative Frequency Histograms (figure A.4.1a);
   (b) Charts: include Conjoint Relative Error charts of each Genome (figure A.4.1b);
   (c) Tables: include Conjoint Kullback-Leibler Tables (figure A.4.1c);
   (d) Data: include text files with Distance Frequency of each symbol by complete genome and by chromosome, and also Conjoint Distance frequency by complete genome and chromosome (figure A.4.1d).
3. You must select the type of data you want to save and choose a directory.
4. You can also save an AGenDA binary file with all data information and results.

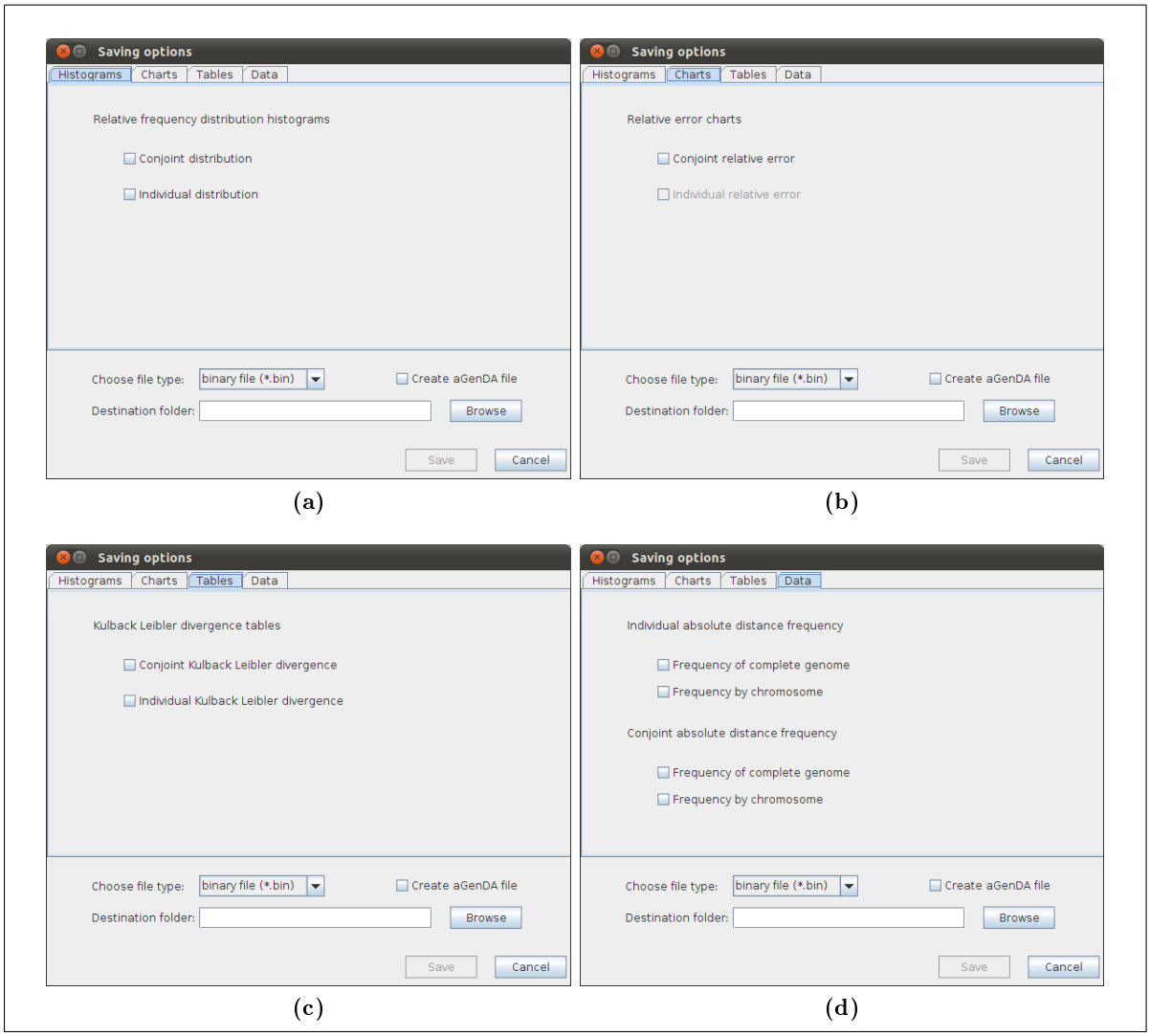


Figure A.4.1: Save window tabs: Histograms tab (a), Charts tab (b), Tables tab menu (c) and Data tab (d).

A.5

Menus Description

A.5.1 File Menu

The File Menu has three submenus:

1. Open FASTA File: loads FASTA files as it was described in section A.2.1.
2. Load AGenDA File: loads a AGenDA binary file as it was described in section A.3.1.
3. Exit: closes the program.

A.5.2 Tools Menu

The Tools Menu has only one submenu: save results. This submenu item permits to save the processed results as it was described in section A.4.1.

A.5.3 Help Menu

The Help Menu is divided in three submenus:

1. Documentation: opens the tool documentation html file (figure A.5.1a).
2. Help Content: opens the tool manual pdf file (figure A.5.1b).
3. About AGenDA: open the about window (figure A.5.1c).

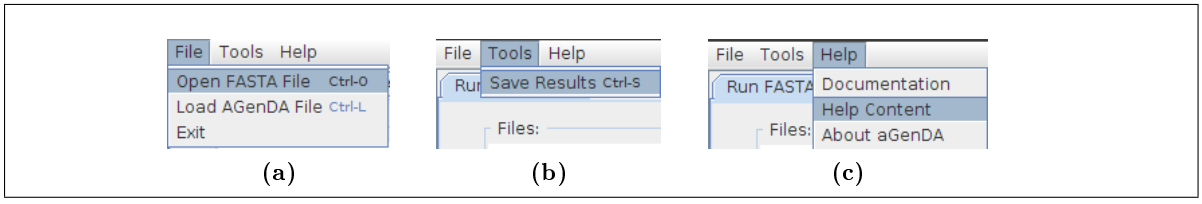


Figure A.5.1: Menu options: file menu (a), tools menu (b) and help menu (c).

Bibliografia

[1] Vera Afreixo, Carlos A. C. Bastos, Armando J. Pinho, Sara P. Garcia, and Paulo J. S. G. Ferreira. Genome analysis with inter-nucleotide distances. *Bioinformatics*, 25(23):3064–3070, December 2009. 2, 14, 15, 16, 32

[2] D. Anastassiou. Evolution of long-range fractal correlations and 1/f noise in DNA base sequences. *IEEE Signal Processing Magazine*, 18:8–20, July 2001. 13

[3] Jim Arlow and Ila Neustadt. *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition) (The Addison-Wesley Object Technology Series).* Addison-Wesley Professional, July 2005. 24, 25

[4] Carlos A. C. Bastos, Vera Afreixo, Armando J. Pinho, Sara P. Garcia, João M. O. S. Rodrigues, and Paulo J. S. G. Ferreira. Distances between dinucleotides in the human genome. In *5th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2011)*, volume 93 of *Advances in Intelligent and Soft Computing*, pages 205–211. Springer Berlin / Heidelberg, 2011. 2, 15, 16, 31, 32

[5] T.A. Brown. *DNA sequencing: the basics.* The Basics Series. Oxford University Press, 1994. 6, 7

[6] C.R. Calladine. *Understanding DNA: the molecule & how it works.* Elsevier Academic Press, 2004. 5

[7] F. S. Collins, E. D. Green, A. E. Guttmacher, and M. S. Guyer. A vision for the future of genomics research. *Nature*, 422(6934):835–847, April 2003. 7

[8] Stefan Kurtz Enno, Stefan Kurtz, Enno Ohlebusch, Chris Schleiermacher, Jens Stoye, and Robert Giegerich. Computation and visualization of degenerate repeats in complete genomes. In *Eight International Symposium on Intelligent Systems for Molecular Biology, La Jolla*, pages 228–238. AAAI Press, 2000. 17

[9] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, and J. M. Merrick. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269(5223):496–512, 1995. 7

[10] Michael Y. Galperin and Guy R. Cochrane. The 2011 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic Acids Research*, 39(suppl 1):D1–D6, January 2011. 19

[11] M. A. Gates. A simple way to look at DNA. *Journal of Theoretical Biology*, 119(3):319–328, 1986. 8

[12] David Gilbert. *The JFreeChart Class Library*. Simba Management Limited, 2002. 29

[13] T. A. Hall. BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. *Nucleic Acids Symposium Series*, 41:95–98, 1999. 18

[14] Eugene Hamori and John Ruskin. H-Curves, A Novel Method of Representation of Nucleotide Series Especially Suited for Long DNA Sequences. *The journal of biological chemistry*, 258(2):1318–1327, 1983. 11, 12

[15] P. He and J. Wang. Numerical characterization of DNA primary sequence. *Internet Electronic Journal of Molecular Design*, 1(12):668–674, 2002. 8

[16] Yu hua Yao and Tian ming Wang. A class of new 2-D graphical representation of DNA sequences and their application. *Chemical Physics Letters*, 398(4-6):318 – 323, 2004. iii, 10, 12

[17] Stefan Kurtz, Jomuna V. Choudhuri, Enno Ohlebusch, Hris C. Schleiermacher, Jens Stoye, and Robert Giegerich. REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucl. Acids. Res.*, 29(22):4633–4642, 2001. 17

[18] P. M. Leong and S. Morgenthaler. Random walk and gap plots of DNA sequences. *Computer Applications in the Biosciences*, 11(5):503–507, 1995. 8

[19] Chun Li and Jun Wang. Numerical characterization and similarity analysis of DNA sequences based on 2-D graphical representation of the characteristic sequences. *Comb Chem High Throughput Screen*, 6(8):795–9, 2003. 10, 11

[20] Elaine R Mardis. Next-generation DNA sequencing methods. *Annual Reviews Genomics Hum Genet*, 9:387–402, 2008. 8

[21] A. M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, 74(2):560–4, 1977. 6

[22] Gregor Mendel. Experiments in Plant Hybridization. *Natural History Society of Brunn in Bohemia*, 4:3–47, 1866. 1

[23] Gabriela Moura, Miguel Pinheiro, Joel Arrais, Ana C. Gomes, Laura Carreto, Adelaide Freitas, José L. Oliveira, and Manuel A. S. Santos. Large Scale Comparative Codon-Pair Context Analysis Unveils General Rules that Fine-Tune Evolution of mRNA Primary Structure. *PLoS ONE*, 2(9):e847, 2007. 17

[24] A.S. Nair and T. Mahalakshmi. Visualisation of genomic data using inter-neucleotide distance signals. Paper presented in the IEEE International Workshop on Genomic Signal Processing, Bucharest, July 2005. 14

[25] Milan Randić, Marjan Vračko, Nella Lerš, and Dejan Plavšić. Analysis of similarity/dissimilarity of DNA sequences based on novel 2-D graphical representatio. *Chemical Physics Letters*, 371(1-2):202 – 207, 2003. 10, 11

[26] Milan Randić, Marjan Vračko, Jure Zupan, and Marjana Novič. Compact 2-D graphical representation of DNA. *Chemical Physics Letters*, 373(5-6):558 – 562, 2003. iii, 11, 12

[27] Milan Randić, Marjan Vračko, Nella Lerš, and Dejan Plavšić. Novel 2-D graphical representation of DNA sequences and their numerical characterization. *Chemical Physics Letters*, 368(1-2):1 – 6, 2003. 10, 11

[28] A. Roy, C. Raychaudhury, and A. Nandy. Novel techniques of graphical representation and analysis of DNA sequences — A review. *Journal of Biosciences*, 23:55–71, 1998. 8

[29] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, C. A. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith. Nucleotide sequence of bacteriophage phi X174 DNA. *Nature*, 265(5596):687–695, February 1977. 6

[30] F. Sanger, A.R. Coulson, G.F. Hong, D.F. Hill, and G.B. Petersen. Nucleotide sequence of bacteriophage phy DNA. *Journal of Molecular Biology*, 162(4):729 – 773, 1982. 7

[31] F. Sanger, S. Nicklen, and A. R. Coulson. DNA Sequencing with Chain-Terminating Inhibitors. *PNAS*, 74(12):5463–5467, 1977. 6

[32] Eric E. Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human molecular genetics*, 19(R2):R227–R240, October 2010. 8

[33] E. Segal, R. Yelensky, A. Kaushal, T. Pham, A. Regev, D. Koller, and N. Friedman. GeneXPress: A Visualization and Statistical Analysis Tool for Gene Expression and Sequence Data. In *Proceedings of the 11th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2004. 18

[34] B. D. Silverman and R. Linsker. A measure of DNA periodicity. *Journal of Theoretical Biology*, 118:295–300, 1986. 14

[35] A. Silverstein and V.B. Silverstein. *DNA*. Science concepts. Twenty-First Century, 2002. 6

[36] Jie Song and Huanwen Tang. A new 2-d graphical representation of dna sequences and their numerical characterization. *Journal of Biochemical and Biophysical Methods*, 63(3):228 – 239, 2005. iii, 9

[37] Jamie Thomas, Daniel Horspool, Gordon Brown, Vasily Tcherepanov, and Chris Upton. GraphDNA: a Java program for graphical display of DNA composition analyses. *BMC Bioinformatics*, 8(1):21+, January 2007. 17

[38] Richard F. Voss. Evolution of long-range fractal correlations and 1/f noise in DNA base sequences. *Physical Review Letters*, 68(25):3805–3808, June 1992. 13

[39] K. Walrath. *The JFC Swing tutorial: a guide to constructing GUIs*. Number vol. 1 in The Java series. Addison-Wesley, 2004. 33

[40] James Watson and Francis Crick's. A struture for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953. 1, 6

[41] Yonghui Wu, Alan Wee-Chung Liew, Hong Yan, and Mengsu Yang. DB-Curve: a novel 2D method of DNA sequence visualization and representation. *Chemical Physics Letters*, 367(1-2):170 – 176, 2003. iii, 9, 10

[42] J. Xiong. *Essential bioinformatics*. Cambridge University Press, 2006. 1

[43] R. Zhang and C. Zhang. Identification of replication origins in archaeal genomes based on the Z-curve method. page 335–346, 2005. iii, 12, 13

[44] Zhu-Jin Zhang. DV-Curve: a novel intuitive tool for visualizing and analyzing DNA sequences. *Bioinformatics*, 25(9):1112–1117, March 2009. iii, 9, 10